LiMo Foundation™

Telephony Foundation API

Version 1.0
March 6, 2008

BETA: Subject to revision prior to final release of LiMo Foundation Platform Release 1

PLEASE REVIEW THE TERMS OF THE FOUNDATION API LICENSE VERSION 1.0 BEFORE READING THIS DOCUMENT.

You may view the Foundation API License v1.0 at:
`http://www.limofoundation.org/images/stories/pdf/limo_foundation_api_license_v1-0.pdf`

This is an agreement between you ("You") and LiMo Foundation related to the use of the APIs contained within this document. By accessing, downloading, reading or otherwise using the APIs contained within this document, You accept the terms of the Foundation API License v1.0. If You do not agree to the terms of this agreement, do not download, read and/or otherwise use any of the APIs.

If You are agreeing to be bound by the Foundation API License v1.0 on behalf of your employer, You represent and warrant that you have full legal authority to bind your employer to this agreement. If You do not have the requisite authority, You may not download, read and/or otherwise use any of the APIs on behalf of your employer.

© Copyright LiMo Foundation 2007, 2008. All rights reserved.

NOTICES

This document and the technical information it contains is subject to change by LiMo Foundation and its members at any time without notice as authorized by LiMo Foundation bylaws and other documents governing LiMo Foundation, including amendments thereto. All statements regarding LiMo's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Any reference in this document to non-LiMo Foundation Web sites do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of this document or any product described in this document and use of those Web sites is at your own risk.

Information concerning non-LiMo products was obtained from the suppliers of those products or a public announcement or source. LiMo cannot confirm the accuracy of that information or the performance, compatibility or any other claims related to non-LiMo products.

All names or other identifying information used in examples in this document are fictitious and any similarity to the names or other identifying information of an actual person or organization is entirely coincidental.

THIS DOCUMENT AND ALL INFORMATION IN THIS DOCUMENT ARE PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OR CONDITION, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY OR CONDITION OF ACCURACY OR COMPLETENESS, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR AGAINST INFRINGEMENT. ALL EXPRESS AND IMPLIED STATUTORY WARRANTIES ARE DISCLAIMED, INCLUDING WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT.

BETA: Subject to revision prior to final release of LiMo Foundation Platform Release 1

© Copyright LiMo Foundation 2007, 2008. All rights reserved.

Contents

1 Telephony Foundation API **1**

1.1 References 1

1.2 Overview 1

1.3 Industry Standard 16

1.4 External API Documents 16

1.5 Plug-in Extension Point Interface 17

1.6 Other Interfaces 17

1.7 Other Notes 17

2 Module Index **19**

2.1 Modules 19

3 Data Structure Index **23**

3.1 Data Structures 23

4 File Index **25**

4.1 File List 25

5 Module Documentation **27**

5.1 CallServices 27

5.2 TelTapiCallSetup 28

5.3 TelTapiCallAnswer 30

5.4 TelTapiCallRelease 32

5.5 TelTapiCallReleaseAll 33

5.6 TelTapiCallSendDTMF 35

5.7 TelTapiCallRetrieve 37

5.8 TelTapiCallHold 38

5.9 TelTapiCallJoin 39

5.10 TelTapiCallSplit 41

5.11 TelTapiCallExplicitTransfer 43

5.12 TelTapiCallGetStatus 45
5.13 TelTapiCallGetDuration 46
5.14 TelTapiCallGetConferenceList 47
5.15 NetTextServices 49
5.16 TelTapiSmsSend 50
5.17 TelTapiSmsRead 52
5.18 TelTapiSmsWrite 54
5.19 TelTapiSmsDelete 56
5.20 TelTapiSmsGetCount 57
5.21 TelTapiSmsCBSetConfig 58
5.22 TelTapiSmsCBGetConfig 60
5.23 TelTapiSmsSetSCA 61
5.24 TelTapiSmsGetSCA 62
5.25 TelTapiSmsSetPreferredBearer 63
5.26 TelTapiSmsGetPreferredBearer 64
5.27 TelTapiSmsGetParameters 65
5.28 TelTapiSmsSetParameters 66
5.29 TelTapiSmsSendDeliveryReport 68
5.30 NetworkServices 69
5.31 TelTapiNwSetConfiguration 70
5.32 TelTapiNwSelectNetwork 72
5.33 TelTapiNwCancelManualSearch 74
5.34 TelTapiNwCancelManualSelection 75
5.35 TelTapiNwGetNetworkInfo 76
5.36 TelTapiNwSearchNetwork 77
5.37 TelTapiNwSetSelectionMode 78
5.38 TelTapiNwGetSelectionMode 80
5.39 TelTapiNwSetServiceDomain 81
5.40 TelTapiNwSetNetworkOrder 83
5.41 TelTapiNwSetNetworkMode 85
5.42 TelTapiNwSetNetworkBand 87
5.43 SIMServices 89
5.44 TelTapiSimGetFileInfo 90
5.45 TelTapiSimReadFile 92
5.46 TelTapiSimUpdateFile 94
5.47 TelTapiSimVerifyPIN 95

5.48 TelTapiSimChangePIN 97
5.49 TelTapiSimDisablePIN 99
5.50 TelTapiSimEnablePIN 101
5.51 TelTapiSimUnblockPIN 102
5.52 TelTapiSimDisableFDN 104
5.53 TelTapiSimEnableFDN 105
5.54 TelTapiSimGetDevCtxdata 106
5.55 TelTapiSimGetIccIdInfo 107
5.56 TelTapiSimGetPINStatus 108
5.57 TelTapiSimGetPIN2Status 109
5.58 TelTapiSimGetFDNStatus 110
5.59 TelTapiSimGetSstInfo 111
5.60 TelTapiSimGetUstInfo 112
5.61 TelTapiSimGetImsiInfo 113
5.62 TelTapiSimGetEccInfo 114
5.63 TelTapiSimGetLanguagePrefInfo 115
5.64 TelTapiSimGetLanguageIndInfo 116
5.65 TelTapiSimGetCardType 117
5.66 TelTapiSimPBGetStorageCount 118
5.67 TelTapiSimPBGetEntryInfo 120
5.68 TelTapiSimPBReadRecord 122
5.69 TelTapiSimPBUpdateRecord 124
5.70 TelTapiSimPBDeleteRecord 125
5.71 SupplementaryServices 127
5.72 TelTapiSsSetCallBarring 128
5.73 TelTapiSsGetCallBarringStatus 130
5.74 TelTapiSsChangeBarPassword 132
5.75 TelTapiSsSetCallForward 134
5.76 TelTapiSsGetCallForwardStatus 136
5.77 TelTapiSsSetCallWaiting 138
5.78 TelTapiSsGetCallWaitingStatus 140
5.79 TelTapiSsVerifyBarPwdResponse 142
5.80 TelTapiSsSetRestrictCallId 144
5.81 TelTapiSsGetRestrictCallId 145
5.82 TelTapiSsQueryCapabilityInfo 146
5.83 TelTapiUssdSendRequest 147

5.84 TelTapiUssdSendUserResponse 149
5.85 Common 151
5.86 TelTapiRegisterClass 157
5.87 TelTapiRegisterEvent 159
5.88 TelTapiDeRegister 161
5.89 TelTapiTunnelReq 163
5.90 TelTapiGetVersionInfo 164
5.91 Call Data Structures 165
5.92 NetText Data Structures 172
5.93 Network Data Structures 177
5.94 SIM Data Structures 182
5.95 SS DataStructures 198
5.96 Utility Data Structures 207

6 Data Structure Documentation 209

6.1 TelCallCugInfo_t Struct Reference 209
6.2 TelCallEndInfo_t Struct Reference 210
6.3 TelCallIncomingCallInfo_t Struct Reference 211
6.4 TelCallingNameInfo_t Struct Reference 213
6.5 TelCallRedirectionInfo_t Struct Reference 214
6.6 TelCallSetupParams_t Struct Reference 215
6.7 TelCallStatus_t Struct Reference 217
6.8 TelCallUserDataInfo_t Struct Reference 218
6.9 TelCallUusInfo_t Struct Reference 219
6.10 TelNetworkConfigSettings_t Struct Reference 220
6.11 TelNetworkInfo_t Struct Reference 221
6.12 TelNetworkList_t Struct Reference 222
6.13 TelNetworkRadioInfo_t Struct Reference 223
6.14 TelNetworkSystemId_t Struct Reference 224
6.15 TelNetworkTimeAndZone_t Struct Reference 226
6.16 TelSimApnData_t Struct Reference 228
6.17 TelSimCallbackEvent_t Struct Reference 229
6.18 TelSimDeviceContext_t Struct Reference 230
6.19 TelSimDeviceContextData_t Struct Reference 231
6.20 TelSimEccData_t Struct Reference 233
6.21 TelSimEccInfo_t Struct Reference 234
6.22 TelSimEvent_t Struct Reference 235

6.23 TelSimFileAccessGeneralResp_t Struct Reference 236
6.24 TelSimFileID_t Struct Reference 237
6.25 TelSimFileInfo_t Struct Reference 238
6.26 TelSimGroupIdentifierInfo_t Struct Reference 240
6.27 TelSimIccIdInfo_t Struct Reference 241
6.28 TelSimImsiInfo_t Struct Reference 242
6.29 TelSimLanguageInfo_t Struct Reference 243
6.30 TelSimPb2gData_t Struct Reference 244
6.31 TelSimPbEntryInfo_t Struct Reference 245
6.32 TelSimPbReadResp_t Struct Reference 246
6.33 TelSimPbRecordData_t Struct Reference 247
6.34 TelSimPbStorageInfo_t Struct Reference 248
6.35 TelSimPbUpdateResp_t Struct Reference 249
6.36 TelSimPinData_t Struct Reference 250
6.37 TelSimPinInfo_t Struct Reference 251
6.38 TelSimReadFileData_t Struct Reference 252
6.39 TelSimReadFileRequestData_t Struct Reference 253
6.40 TelSimServiceTable_t Struct Reference 255
6.41 TelSimSstInfo_t Struct Reference 261
6.42 TelSimStatusData_t Struct Reference 262
6.43 TelSimUpdateFileRequestData_t Struct Reference 264
6.44 TelSimUstInfo_t Struct Reference 266
6.45 TelSmsAddressInfo_t Struct Reference 267
6.46 TelSmsCbConfig_t Struct Reference 268
6.47 TelSmsCbMsg_t Struct Reference 269
6.48 TelSmsData_t Struct Reference 270
6.49 TelSmsDatapackageInfo_t Struct Reference 271
6.50 TelSmsParams_t Struct Reference 272
6.51 TelSmsStoredMsgCountInfo_t Struct Reference 274
6.52 TelSsCallBarringInfo_t Struct Reference 275
6.53 TelSsCallBarringRecord_t Struct Reference 276
6.54 TelSsCallWaitingRecord_t Struct Reference 277
6.55 TelSsCapabilityInfo_t Struct Reference 278
6.56 TelSsCliRecord_t Struct Reference 279
6.57 TelSsForwardInfo_t Struct Reference 280
6.58 TelSsForwardRecord_t Struct Reference 281

6.59 TelSsInfo_t Struct Reference 282
6.60 TelSsRecord_t Union Reference 283
6.61 TelSsUssdMsgInfo_t Struct Reference 284
6.62 TelTapiErrorInfo_t Struct Reference 285
6.63 TelTapiEvent_t Struct Reference 286
6.64 TelTapiTunnelInfo_t Struct Reference 287
6.65 TelTapiVersionInfo_t Struct Reference 288
6.66 TelUsimEccInfo_t Struct Reference 289
6.67 TelUSimServiceTable_t Struct Reference 290

7 File Documentation 297

7.1 ComponentOverview.h File Reference 297
7.2 ITapiCall.h File Reference 298
7.3 ITapiNetText.h File Reference 300
7.4 ITapiNetwork.h File Reference 302
7.5 ITapiSim.h File Reference 304
7.6 ITapiSs.h File Reference 307
7.7 TapiCommon.h File Reference 309
7.8 TelCall.h File Reference 312
7.9 TelNetText.h File Reference 315
7.10 TelNetwork.h File Reference 317
7.11 TelSim.h File Reference 319
7.12 TelSs.h File Reference 324
7.13 TelUtility.h File Reference 327

Chapter 1

Telephony Foundation API

1.1 References

[1] *Software Architecture Document*, Version 1.1 Revision 1.4, LiMo Foundation, 11 September 2007

[2] *High Level Requirements*, Version 1.0 Revision 1.5, LiMo Foundation, 22 October 2007

[3] 3GPP Specification References:

- 3GPP TS 24.008 Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 5)
- 3GPP TS 22.030 Man-Machine Interface (MMI) of the User Equipment (UE) (Release 5)
- 3GPP TS 24.080 supplementary services specification (Release 5)
- 3GPP TS 23.038 Alphabets and language-specific information (Release 5)
- 3GPP TS 23.040 Technical realization of the Short Message Service (SMS) (Release 5)
- 3GPP TS 29.002 Mobile Application Part (MAP) specification (Release 5)
- 3GPP TS 31.102 Characteristics of the USIM application (Release 5)
- 3GPP TS 22.084 MultiParty (MPTY) Supplementary Services - Stage 1 (Release 5)
- 3GPP TS 23.014 Support of Dual Tone Multi Frequency (DTMF) signaling (Release 5)

1.2 Overview

The Telephony Service provides an abstraction of the mobile network telephony capabilities of a device and enables applications to invoke comprehensive and flexible telephony functions in support of their applications; e.g. voice call services, video call setup (video call functionality is not supported in the current release), supplementary services, call charge information and telephone network registration. The Telephony service also provides the telephony related services required by other LiMo Frameworks and components, for example APIs for sending/receiving SMS messages, receiving cell broadcast messages for the LiMo Messaging Service.

SIM/UIM access (e.g. read/write of SIM resident data) is closely related to telephony functionality and often realized by the same components and therefore this is also included in the Telephony Service architecture. SIM/USIM Application Toolkit provides a mechanism that allows applications resident on the

SIM/USIM to interact with the mobile handset applications and components. Support of this functionality is not a part of LiMo Release 1 scope, but included as a placeholder for a future release.

1.2.1 Execution Model - TAPI Client Library and Telephony Server

The LiMo TAPI client library and LiMo Telephony Server provide the general telephony services and hide all low-level details from the application developers.

1) A Telephony client application loads the LiMo TAPI Client library into its process space and uses TAPI to interface with LiMo Telephony Server.

2) TAPI client application has to register the callback with TAPI Library for handling the telephony events (Solicited /Unsolicited events). Registered callback function shall be invoked when events occurs at TAPI Library.

3) LiMo TAPI library establishes the communication link with the LiMo Telephony Server. When an application calls a Telephony APIs, the LiMo TAPI library validates and marshals the parameters, then forwards the information to the LiMo Telephony Server.

4) LiMo Telephony Server tracks communications resources available and sends their synchronous/asynchronous responses to LiMo TAPI library. TAPI library passes the synchronous/asynchronous responses or any unsolicited indication in terms of Telephony events to TAPI Application directly hiding telephony protocol details.

5) LiMo TAPI library de-marshals the response from LiMo Telephony Server which is returned synchronously and returns it to LiMo TAPI Application.

There is only one Telephony Server instance in LiMo R1 which supports multiple simultaneous telephony applications. LiMo Telephony frameworks client-server model abstracts the telephony applications from protocol and low-level details, which make simple for application developer and helps telephony server robust in nature On the other hand, LiMo TAPI already supports radio interfaces GSM/CDMA/3G and it is easy to extend to support various other radio interfaces.

1.2.2 General Programming Considerations - TAPI Request APIs

The TAPI library provides APIs for applications to access telephony services. The TAPI library provides two types of request functions. These request functions are called by the telephony client applications in order to provide basic telephony service to the user.

- Synchronous requests
- Asynchronous requests

1.2.2.1 Synchronous Request APIs

The client application process waits until the processing of the request is completed. The Telephony server can process some APIs and provide the return response while returning back to the function. The function returns with the result of the request (if any) and any applicable error codes. For example, an application may query for call status information on a call or get network registration status information.

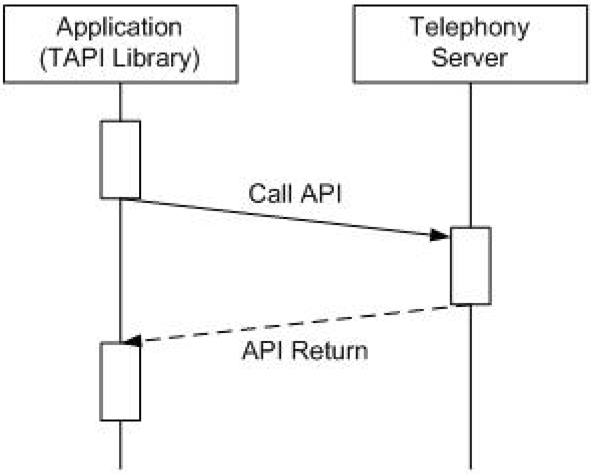


Figure 1.1: Synchronous Interface

1.2.2.2 Asynchronous Request APIs

In the case of Asynchronous request APIs, the request is provided to the Telephony server and the actual confirmation/response to that request will be returned at later time. The Telephony server provides a response to the TAPI library and the TAPI library invokes registered application callback to pass telephony events to the application. An illustration of the client application making a call to an asynchronous function is provided in the following figure.

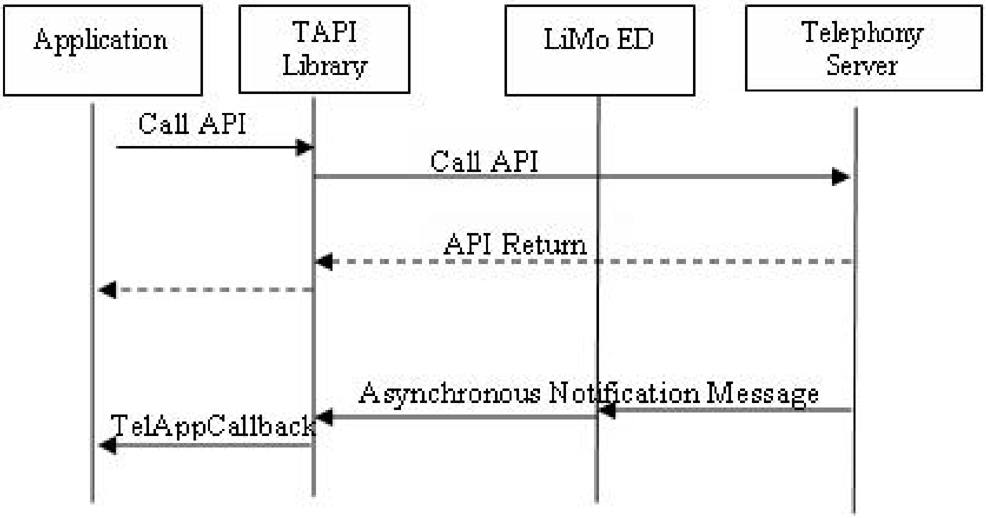


Figure 1.2: Asynchronous Interface

In an asynchronous request:

- The client application passes parameters related to the service request and also it provides an out parameter request id (an unique identifier) for each asynchronous service request which will be filled and returned by Telephony server.

- When the function is returned, the return value indicates whether the request was successfully sent to the Telephony Server or if failure happened during processing in the Telephony Server. In case of a successful API return, a valid RequestId value will be filled and returned to the application. The request id parameter value will be set to an invalid value (for e.g., 0xFF), when the API fails due to some reason.

- When the Telephony server receives the actual completion of service, it sends the results of the events to the TAPI library. The TAPI library processes the event received and passes it to Client application using the registered callback. The client application shall handle the notification events in the registered callback.

- Asynchronous functions can be completed in one or more event notifications.

For example, In a Call Hold API request, there will be one event notification which will complete the asynchronous nature.

However, in a Call Setup API request, there are many intermediate events during the call including: a Connecting Indication event, an Alert Indication event, and a Connected Indication event. Based on the event notification and corresponding states, the application can decide whether the request is complete or not.

The Telephony client application has to register the callback with Telephony to receive the event notification messages and handle them in its own process. Based on the callback registration, the application callback will get invoked for those registered events when the TAPI library receives the indication from Telephony server. For example, a call being connected, Alert Indication, or incoming call etc.

If an application decides to call an asynchronous API and then wait for the API completion notification, then the application can be considered to be executing that API in a synchronous fashion. This is allowable.

RequestId Information:

The request id is used to uniquely identify an asynchronous service request. All Asynchronous Request APIs provide a RequestId as an out parameter. The RequestId pointer will have a value in the range(0-255) returned by Telephony server on processing the API request successfully. If there is a failure in an API request, the RequestId shall be invalid request id (value = -1) and the function return value will indicate an error.
This RequestId will act as reference for each API request an application originates. An application shall receive the RequestId value along with other event data for asynchronous response/Indication from TAPI library. The application can clear the RequestId value based on the RequestId and event(s) received for the originated request. In the case of unsolicited events, the RequestId parameter passed in the notification shall be the invalid request id value which is -1.

1.2.2.3 TAPI Considerations

The following conditions need to be followed by TAPI and Applications:

1. All the input parameters passed in Telephony APIs are validated in TAPI Client library as below:
   (a) Parameters (passed by reference) are validated against Null
   (b) Type defined Enumeration parameters are validated against start and end enumeration values

   (c) TAPI Client library will pass all the requests to Telephony Server after the above considerations (a) and (b).
   (d) Telephony server shall validate some parameters based on the information. For example, if an invalid call handle is sent in Call API requests, the Telephony server will validate the call handle with its existing call related information. If it is not a valid handle, then a TAPI_INVALID_-CALLHANDLE error will be returned from API synchronously in the return value to the client application.

2. The Application has to allocate memory for the parameters passed by reference in Telephony API calls. The Application shall de-allocate the memory once the API call is completed, successful or not.

3. In the case of Synchronous APIs, when API has returned with success to the Application, then the out parameters' data is valid. Application can process the out parameter data and release if any memory is allocated for that out parameter.

4. In the case of Asynchronous APIs, data will be available as soon as application callback is invoked for a received telephony event from TAPI library.

5. When an API is processed successfully within Telephony FW, the API return value will be set to TAPI_API_SUCCESS. If there is a failure in processing the request within Telephony server, then the API return value may contain any other value than TAPI_API_SUCCESS and the error values are defined in section [TapiResult_t]. For synchronous APIs out parameters' data is valid when the API return value is set to TAPI_API_SUCCESS.

6. If there are errors in processing the request at the OEM Telephony provider or any errors from the network, these errors are indicated to the application in Asynchronous notification based on the service request.

7. If there is no data associated with an event, then the event data parameter in the TAPI Event structure [TelTapiEvent_t] will be NULL when it is passed in the Application callback.

8. The first TAPI request from a TAPI Client will initialize the TAPI Library and Telephony server communication.

9. In the case of asynchronous solicited responses, TAPI library filters out the event with request id which is not requested by that application.

10. The API request may fail with a return error TAPI_API_REQUEST_MAX_IN_PROGRESS as the maximum number of requests for the same service is already in progress in TS/OEM Adaptation layer plug-in.

11. If an application calls a synchronous API, then it needs to wait until the request is complete (blocked) and the response is sent back. But at the same time, The application should be able to process the registered callbacks, if any, it has registered for other events.

1.2.2.4 The Two Common Uses of Asynchronous Requests

In general, there are 2 cases for Asynchronous interaction between a TAPI application and the Telephony Server.

Case1 (Service Requests): A TAPI Application requests the service by calling a TAPI Client Library function and then the TAPI Library sends the notification to inform the application of the service result by invoking the registered application callback. Making a voice call is such a case.

The application calls TelTapiCallSetup to request the voice call service, and then it may continue with other work. The Telephony server will provide intermediate notifications (Setup indication, Connecting Indication, Alert indication etc.) about any changes in call status. Once the call is established, the Telephony server will send the Connected indication to the TAPI Library which will in turn notify the application about the status of call (connected or disconnected), the application should handle the processing after that point.

Case2 (Unsolicited Indications): The Telephony server on receiving some unsolicited indications which may be due to messages received from the network or from an underlying wireless protocol stack performing some periodic functionalities like network search and registration, will inform these indications to the application through asynchronous notifications using the TAPI Library. Examples of this case are incoming SMS, incoming call, network registration status and network signal change.

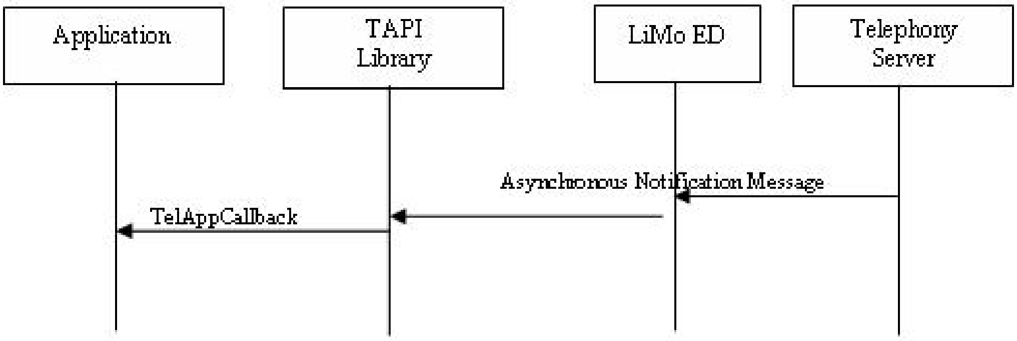


Figure 1.3: Notification Case2 Illustration

An application has to register the callback for Telephony events with the TAPI Library in order to receive the solicited or unsolicited asynchronous notification. Unsolicited notification (case 2) will occur regardless whether application uses only synchronous API or only asynchronous API or combination of those. All applications should be able to handle asynchronous unsolicited events.

1.2.3 Unsolicited Notifications

When an unsolicited notification is generated by Telephony FW in response to some external event (e.g. a signal), it does not directly mean completion or progress of any outstanding request posted by a client application. An unsolicited event in general means either change of some state (for example, network registration change) or some incoming signal that may be of interest for the application (e.g. an incoming call).

An unsolicited asynchronous notification still has to be handled even if an application uses only synchronous API. An application may choose to ignore unsolicited notifications, however TAPI does not provide any other direct way for an application to be informed on telephony related external events. The list of unsolicited notifications supported by the Telephony server is captured in Overview2_7 .

1.2.4 Event Notifications

Telephony applications (i.e. clients) have to register for events which are of its interest according to its requirements with the TAPI Library. On receiving an asynchronous response or unsolicited indication, the TAPI Library will invoke the callback registered by the telephony client. TAPI event details are detailed in TelTapiEvent_t .

Applications will register the callback with TAPI Library for Telephony events TAPI library shall invoke the registered callback function when the event is received. See the complete list of Telephony events in Overview2_6.

1.2.5 The TAPI Event Class

Please refer to TelTapiEventClass_t for a complete description of this enumerated list of events.

Event Class	**Description**
TAPI_CALL_VOICE	Voice call events
TAPI_CALL_DATA	Data call events
TAPI_NETTEXT	Network Text (SMS and Cell Broadcast) related events
TAPI_NETWORK	Network related events
TAPI_SS	Supplementary Service events
TAPI_SIM	Sim related events
TAPI_ERR	Error Related events (For debugging purposes e.g. Fatal errors at OEM level to be displayed on screen)
TAPI_UTIL	Utility events
TAPI_ALL	Events pertaining to All Telephony Event Classes

1.2.6 Telephony Events

This table explains the list of TAPI events which are passed to an application based on confirmation/indication event occurring at the TAPI Library.

Network Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_NETWORK_SELECT_CNF	This confirmation event will be returned when user manually selected a network from list for Network selection	TelNetworkInfo_t	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_SEARCH_CNF	This event is returned as confirmation for manual Network search with the available network list	TelNetworkList_t	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF	This event is returned when Get selection mode indication is received at TS	TelNetworkSelectionMode_t	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF	This event is returned when Get network information is requested	TelNetworkInfo_t	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF	This event is returned on cancellation of manual network search	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF	This event is returned on cancellation of manual network selection from network list	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_SETNWORDER_CNF	This event is returned on setting network order	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_SETNWMODE_CNF	This event is returned on setting network mode	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF	This event is returned on setting network selection mode	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_SETNWBAND_CNF	This event is returned on setting network band	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_SETSERVICEDOMAIN_CNF	This event is returned on setting network service domain	No Data associated with this event	TelNetworkOperationCause_t
TAPI_EVENT_NETWORK_CONFIG_CNF	This event is returned on setting the network configuration	No Data associated with this event	TelNetworkOperationCause_t

TAPI_EVENT_NETWORK_CHANGE_IND	This event will be returned when Telephony server receives any indication from OEM about change in Network registration/selection/status	TelNetworkInfo_t	No Status associated with this event
TAPI_EVENT_NETWORK_SEARCH_IND	This is unsolicited event indicated when TS receives any indication of Manual search indication with list	TelNetworkList_t	No Status associated with this event
TAPI_EVENT_NETWORK_RADIO_INFO_IND	This is unsolicited event indicating Radio signal indication	TelNetworkRadioInfo_t	No Status associated with this event

NetText Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_NETTEXT_READ_SMS_CNF	This event is returned in response to SMS Read message	TelSmsData_t	TelSmsResponse_t
TAPI_EVENT_NETTEXT_GET_COUNT_CNF	This event is returned in response to Get Number of messages stored in SIM	TelSmsStoredMsgCountInfo_t	TelSmsResponse_t
TAPI_EVENT_NETTEXT_SENTSTATUS_CNF	This event is returned in response to SMS message sent	TelSmsDatapackageInfo_t	TelSmsResponse_t
TAPI_EVENT_NETTEXT_GET_PARAM_CNF	This event is returned in response to NetText Get parameters	TelSmsParams_t	TelSmsResponse_t
TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF	This confirmation event is a returned indicating the delete message e status	Integer:Sim-index from which the message has been deleted	TelSmsResponse_t
TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF	This confirmation event is a returned indicating the Save SMS into SIM status	Integer:Sim-Index in which the message stored	TelSmsResponse_t
TAPI_EVENT_NETTEXT_SET_REQUEST_CNF	This event will be returned as a confirmation for SET nettext APIs	No Data associated with this event	TelSmsResponse_t
TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF	This event will be returned with Get CB configuration details	TelSmsCbConfig_t	TelSmsResponse_t
TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF	This event is returned as a response to get preferred bearer used for SMS Transport	TelSmsBearerType_t	TelSmsResponse_t

TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF	This event is returned in response to send delivery report	No Event data associated with this event	TelSmsResponse_t
TAPI_EVENT_NETTEXT_INCOM_IND	This is an unsolicited Incoming SMS Event	TelSmsDatapackageInfo_t	No Status associated with this event
TAPI_EVENT_NETTEXT_CB_INCOM_IND	This is an unsolicited Incoming cell broadcast message	TelSmsCbMsg_t	No Status associated with this event
TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND	This event is an unsolicited event indicating SIM memory status	BOOL (SIM Memory is available or not : 1:0)	No Status associated with this event

Call Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_CALL_SETUP_CNF	This event is returned as confirmation for Call setup	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_SEND_DTMF_CNF	This event indicates the confirmation for sending DTMF Digits to the network	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_ANSWER_CNF	Incoming call was answered	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_RETRIEVE_CNF	This event indicates Call is retrieved from held state	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_HOLD_CNF	This event indicates is call is put on hold	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_TRANSFER_CNF	This event indicates is call is explicitly transferred	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_SETUPCONFERENCE_CNF	This event indicates that Call is joined in a conference	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_SPLITCONFERENCE_CNF	This event indicates Call is split from conference	Integer: Call Handle	TelCallCause_t
TAPI_EVENT_CALL_FORWARD_IND	This event indication for Call is forwarded	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_ALERT_IND	This event indicates the Call is in Alerting stage	Integer: Call Handle	No Status associated with this event

TAPI_EVENT_CALL_RELEASE_ALL_CNF	This event is indicated when all the calls released requested by the application	No Data associated with this event	TelCallCause_t
TAPI_EVENT_CALL_CONNECTED_IND	This event indicates the Call is Connected	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_CONNECTING_IND	This event indicates the Call is in Connecting stage	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_RETRIEVE_IND	This event indicates that other party has retrieved the call from hold	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_HOLD_IND	This event indicates that other party is has put the call on hold	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_TRANSFER_IND	This event is to indicate the notification of remote Explicit Call Transfer	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_SETUPCONFERENCE_IND	This event is to indicate the notification of invocation of Multiparty indication	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_WAITING_IND	This event is to indicate that Call waiting indication at the remote end	Integer: Call Handle	No Status associated with this event
TAPI_EVENT_CALL_USERINFO_IND	This event is indicated when user-User Information(UUS1) exchanged during the call	TelCallUserDataInfo_t	No Status associated with this event
TAPI_EVENT_CALL_INCOM_IND	This event indicates Incoming Call indication to Application	TelCallIncomingCallInfo_t	No Status associated with this event
TAPI_EVENT_CALL_END_IND	This event indicates the Call is Ended. This event can be solicited/unsolicited. In case of solicited events (call release/end is explicitly requested by Application) then valid request id shall be present. In case of Unsolicited events, invalid (-1) is passed in the event notification	TelCallEndInfo_t	No Status associated with this event

Supplementary Service Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_SS_BARRING_CNF	This confirmation event is sent when Barring activation/deactivation is requested	TelSsInfo_t	TelSsCause_t
TAPI_EVENT_SS_GET_PASSWORD_IND	This event is indicated to the Application when network requests for Call barring Get password for barring operation.	TelSsBarPwdType_t	No Status associated with this event
TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF	This event is returned as Barring password change confirmation	No Data associated with this event	TelSsCause_t
TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF	This event is returned as Barring status query confirmation	TelSsInfo_t	TelSsCause_t
TAPI_EVENT_SS_FORWARD_CNF	This event is returned as confirmation for Forward Activation/Deactivation/Registration/Erasure	TelSsInfo_t	TelSsCause_t
TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF	This event is returned as confirmation for Forward Status query confirmation	TelSsInfo_t	TelSsCause_t
TAPI_EVENT_SS_WAITING_CNF	This event is returned as confirmation for Call waiting activation/deactivation	TelSsInfo_t	TelSsCause_t
TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF	This event is returned as Call waiting status query confirmation	TelSsInfo_t	TelSsCause_t
TAPI_EVENT_SS_USSD_CNF	This event is returned as USSD send confirmation	TelSsUssdMsgInfo_t	TelSsCause_t
TAPI_EVENT_SS_USSDCNF_RSP	This event is returned as USSD confirmation response which is sent as a response to received USSD	TelSsUssdRespType_t	TelSsCause_t
TAPI_EVENT_SS_USSDCNF_NOTIFY	This event is an unsolicited USSD notification	TelSsUssdMsgInfo_t	TelSsCause_t
TAPI_EVENT_SS_USSD_IND_RSP	USSD Indication Response	TelSsUssdRespType_t	TelSsCause_t

TAPI_EVENT_SS_-VERIFY_PWD_CNF	This event is sent when verify pwd response is sent to Network	No Data associated with this event	TelSsCause_t
TAPI_EVENT_SS_-QUERYCAPABILITY_-INFO_CNF	Event associated with the interrogation of supported supplementary services at OEM.	TelSsCapabilityInfo_t	TelSsCause_t
TAPI_EVENT_SS_-USSD_IND	This is an unsolicited USSD indication	TelSsUssdMsgInfo_t	No Status associated with this event
TAPI_EVENT_SS_-USSD_IND_NOTIFY	This event is an unsolicited USSD notification	TelSsInfo_t	No Status associated with this event

SIM Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_SIM_FILE_INFO_CNF	SIM elementary file information is returned	TelSimFileInfo_t	TelSimAccessResult_t
TAPI_EVENT_SIM_READ_FILE_CNF	This event is returned along with Read file information	TelSimReadFileData_t	TelSimAccessResult_t
TAPI_EVENT_SIM_UPDATE_FILE_CNF	This event is returned along with Updated file information	TelSimFileAccessGeneralResp_t	TelSimAccessResult_t
TAPI_EVENT_SIM_VERIFY_PIN_CNF	This confirmation event is returned when PIN is verified	TelSimPinInfo_t	TelSimPinOperationResult_t
TAPI_EVENT_SIM_CHANGE_PIN_CNF	This confirmation event is returned when PIN is changed	TelSimPinInfo_t	TelSimPinOperationResult_t
TAPI_EVENT_SIM_DISABLE_PIN_CNF	This confirmation event is returned when PIN is disabled	TelSimPinInfo_t	TelSimPinOperationResult_t
TAPI_EVENT_SIM_ENABLE_PIN_CNF	This confirmation event is returned when PIN is enabled	TelSimPinInfo_t	TelSimPinOperationResult_t
TAPI_EVENT_SIM_UNBLOCK_PIN_CNF	This confirmation event is returned on unblock PIN request	TelSimPinInfo_t	TelSimPinOperationResult_t
TAPI_EVENT_SIM_ENABLE_FDN_CNF	This event will be returned when FDN is enabled	No Data associated with this event	TelSimPinOperationResult_t
TAPI_EVENT_SIM_DISABLE_FDN_CNF	This event will be returned when FDN is disabled	No Data associated with this event	TelSimPinOperationResult_t
TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF	This event is notified for 2G Phone book storage count query	TelSimPbStorageInfo_t	TelSimPbAccessResult_t
TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF	Phone book entry information	TelSimPbEntryInfo_t	TelSimPbAccessResult_t
TAPI_EVENT_SIM_PB_ACCESS_READ_CNF	This event is returned when phone book entry is read	TelSimPbReadResp_t	TelSimPbAccessResult_t
TAPI_EVENT_SIM_PB_UPDATE_CNF	This event is returned once phonebook update is completed	TelSimPbUpdateResp_t	TelSimPbAccessResult_t
TAPI_EVENT_SIM_PB_DELETE_CNF	This event is returned once phonebook record is deleted	TelSimPbUpdateResp_t	TelSimPbAccessResult_t

TAPI_EVENT_SIM_STATUS_IND	This event is sent whenever there is a change in the SIM status. Application can call GetPinStatus/GetEccInfo/GetICCIDInfo APIs based on SIM state. For example, if Sim state is PIN required, application can call getpinstatus to know whether pin/upin is enabled/blocked.	TelSimCardStatus_t	No Status associated with this event
TAPI_EVENT_ SIM_FDN_STATUS_IND	This event is to indicate the whenever FDN status is changed(disabled/enabled)	Bool fdn_status. (True: enabled, False:disabled)	No Status associated with this event

Error Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_ERR_RESERVED1_IND	This event is notified to application about some error has occurred at OEM and can be used for debugging purpose	TelTapiErrorInfo_t	No Status associated with this event
TAPI_EVENT_ERR_RESERVED2_IND	This event is notified to application about some error has occurred at OEM and can be used for debugging purpose	TelTapiErrorInfo_t	No Status associated with this event
TAPI_EVENT_ERR_RESERVED3_IND	This event is notified to application about some error has occurred at OEM and can be used for debugging purpose	TelTapiErrorInfo_t	No Status associated with this event
TAPI_EVENT_ERR_RESERVED4_IND	This event is notified to application about some error has occurred at OEM and can be used for debugging purpose	TelTapiErrorInfo_t	No Status associated with this event
TAPI_EVENT_ERR_RESERVED5_IND	This event is notified to application about some error has occurred at OEM and can be used for debugging purpose	TelTapiErrorInfo_t	No Status associated with this event

UTILITY Class Events			
TAPI Event Type	**Description**	**Associated Event Data/Payload Structures**	**Asynchronous Response Status**
TAPI_EVENT_TUNNEL_INFO_CNF	This confirmation event is returned for Tunnel information	TelTapiTunnelInfo_t	Tunnel status is a Boolean. Success (1-True) or Failure (0-False) indicates asynchronous return
TAPI_EVENT_TUNNEL_INFO_IND	This event is an unsolicited event informing about Tunnel information	TelTapiTunnelInfo_t	No Status associated with this event

1.2.7 Mapping of LiMo Event Ids and Unsolicited TAPI Events

The table below indicates the unsolicited TAPI events and their corresponding LiMo event ids. Some of these event ids can be used for auto-launch purposes.

LiMo Telephony Event Ids	**TAPI Event**
Telephony.Network.ChangeInd	TAPI_EVENT_NETWORK_CHANGE_IND
Telephony.Network.SearchInd	TAPI_EVENT_NETWORK_SEARCH_IND
Telephony.Network.RadioinfoInd	TAPI_EVENT_NETWORK_RADIO_INFO_IND
Telephony.Nettext.IncomingInd	TAPI_EVENT_NETTEXT_INCOM_IND
Telephony.Nettext.CbIncomingInd	TAPI_EVENT_NETTEXT_CB_INCOM_IND
Telephony.Nettext.MemoryStatusInd	TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND
Telephony.Call.Voice.IncomingInd	TAPI_EVENT_CALL_INCOM_IND
Telephony.Ss.UssdInd	TAPI_EVENT_SS_USSD_IND
Telephony.Ss.UssdIndNotify	TAPI_EVENT_SS_USSD_IND_NOTIFY
Telephony.Sim.SimStatusInd	TAPI_EVENT_SIM_STATUS_IND
Telephony.Sim.FdnStatusInd	TAPI_EVENT_SIM_FDN_STATUS_IND
Telephony.Err.ErrInd1	TAPI_EVENT_ERR_RESERVED1_IND
Telephony.Err.ErrInd2	TAPI_EVENT_ERR_RESERVED2_IND
Telephony.Err.ErrInd3	TAPI_EVENT_ERR_RESERVED3_IND
Telephony.Err.ErrInd4	TAPI_EVENT_ERR_RESERVED4_IND
Telephony.Err.ErrInd5	TAPI_EVENT_ERR_RESERVED5_IND
Telephony.Util.TunnelInfoInd	TAPI_EVENT_TUNNEL_INFO_IND

1.3 Industry Standard

None

1.4 External API Documents

None

1.5 Plug-in Extension Point Interface

None

1.6 Other Interfaces

None

1.7 Other Notes

None

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

CallServices 27
  TelTapiCallSetup 28
  TelTapiCallAnswer 30
  TelTapiCallRelease 32
  TelTapiCallReleaseAll 33
  TelTapiCallSendDTMF 35
  TelTapiCallRetrieve 37
  TelTapiCallHold 38
  TelTapiCallJoin 39
  TelTapiCallSplit 41
  TelTapiCallExplicitTransfer 43
  TelTapiCallGetStatus 45
  TelTapiCallGetDuration 46
  TelTapiCallGetConferenceList 47
NetTextServices 49
  TelTapiSmsSend 50
  TelTapiSmsRead 52
  TelTapiSmsWrite 54
  TelTapiSmsDelete 56
  TelTapiSmsGetCount 57
  TelTapiSmsCBSetConfig 58
  TelTapiSmsCBGetConfig 60
  TelTapiSmsSetSCA 61
  TelTapiSmsGetSCA 62
  TelTapiSmsSetPreferredBearer 63
  TelTapiSmsGetPreferredBearer 64
  TelTapiSmsGetParameters 65
  TelTapiSmsSetParameters 66
  TelTapiSmsSendDeliveryReport 68
NetworkServices 69
  TelTapiNwSetConfiguration 70
  TelTapiNwSelectNetwork 72

TelTapiNwCancelManualSearch 74
TelTapiNwCancelManualSelection 75
TelTapiNwGetNetworkInfo 76
TelTapiNwSearchNetwork 77
TelTapiNwSetSelectionMode 78
TelTapiNwGetSelectionMode 80
TelTapiNwSetServiceDomain 81
TelTapiNwSetNetworkOrder 83
TelTapiNwSetNetworkMode 85
TelTapiNwSetNetworkBand 87

SIMServices 89

TelTapiSimGetFileInfo 90
TelTapiSimReadFile 92
TelTapiSimUpdateFile 94
TelTapiSimVerifyPIN 95
TelTapiSimChangePIN 97
TelTapiSimDisablePIN 99
TelTapiSimEnablePIN 101
TelTapiSimUnblockPIN 102
TelTapiSimDisableFDN 104
TelTapiSimEnableFDN 105
TelTapiSimGetDevCtxdata 106
TelTapiSimGetIccIdInfo 107
TelTapiSimGetPINStatus 108
TelTapiSimGetPIN2Status 109
TelTapiSimGetFDNStatus 110
TelTapiSimGetSstInfo 111
TelTapiSimGetUstInfo 112
TelTapiSimGetImsiInfo 113
TelTapiSimGetEccInfo 114
TelTapiSimGetLanguagePrefInfo 115
TelTapiSimGetLanguageIndInfo 116
TelTapiSimGetCardType 117
TelTapiSimPBGetStorageCount 118
TelTapiSimPBGetEntryInfo 120
TelTapiSimPBReadRecord 122
TelTapiSimPBUpdateRecord 124
TelTapiSimPBDeleteRecord 125

SupplementaryServices 127

TelTapiSsSetCallBarring 128
TelTapiSsGetCallBarringStatus 130
TelTapiSsChangeBarPassword 132
TelTapiSsSetCallForward 134
TelTapiSsGetCallForwardStatus 136
TelTapiSsSetCallWaiting 138
TelTapiSsGetCallWaitingStatus 140
TelTapiSsVerifyBarPwdResponse 142
TelTapiSsSetRestrictCallId 144
TelTapiSsGetRestrictCallId 145
TelTapiSsQueryCapabilityInfo 146
TelTapiUssdSendRequest 147
TelTapiUssdSendUserResponse 149

Common 151

  - TelTapiRegisterClass 157
  - TelTapiRegisterEvent 159
  - TelTapiDeRegister 161
  - TelTapiTunnelReq 163
  - TelTapiGetVersionInfo 164
- Call Data Structures 165
- NetText Data Structures 172
- Network Data Structures 177
- SIM Data Structures 182
- SS DataStructures 198
- Utility Data Structures 207

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

TelCallCugInfo_t . . . 209
TelCallEndInfo_t . . . 210
TelCallIncomingCallInfo_t . . . 211
TelCallingNameInfo_t . . . 213
TelCallRedirectionInfo_t . . . 214
TelCallSetupParams_t . . . 215
TelCallStatus_t . . . 217
TelCallUserDataInfo_t . . . 218
TelCallUusInfo_t . . . 219
TelNetworkConfigSettings_t . . . 220
TelNetworkInfo_t . . . 221
TelNetworkList_t . . . 222
TelNetworkRadioInfo_t . . . 223
TelNetworkSystemId_t . . . 224
TelNetworkTimeAndZone_t . . . 226
TelSimApnData_t . . . 228
TelSimCallbackEvent_t . . . 229
TelSimDeviceContext_t . . . 230
TelSimDeviceContextData_t . . . 231
TelSimEccData_t . . . 233
TelSimEccInfo_t . . . 234
TelSimEvent_t . . . 235
TelSimFileAccessGeneralResp_t . . . 236
TelSimFileID_t . . . 237
TelSimFileInfo_t . . . 238
TelSimGroupIdentifierInfo_t . . . 240
TelSimIccIdInfo_t . . . 241
TelSimImsiInfo_t . . . 242
TelSimLanguageInfo_t . . . 243
TelSimPb2gData_t . . . 244
TelSimPbEntryInfo_t . . . 245
TelSimPbReadResp_t . . . 246
TelSimPbRecordData_t . . . 247

TelSimPbStorageInfo_t 248
TelSimPbUpdateResp_t 249
TelSimPinData_t 250
TelSimPinInfo_t 251
TelSimReadFileData_t 252
TelSimReadFileRequestData_t 253
TelSimServiceTable_t 255
TelSimSstInfo_t 261
TelSimStatusData_t 262
TelSimUpdateFileRequestData_t 264
TelSimUstInfo_t 266
TelSmsAddressInfo_t 267
TelSmsCbConfig_t 268
TelSmsCbMsg_t 269
TelSmsData_t 270
TelSmsDatapackageInfo_t 271
TelSmsParams_t 272
TelSmsStoredMsgCountInfo_t 274
TelSsCallBarringInfo_t 275
TelSsCallBarringRecord_t 276
TelSsCallWaitingRecord_t 277
TelSsCapabilityInfo_t 278
TelSsCliRecord_t 279
TelSsForwardInfo_t 280
TelSsForwardRecord_t 281
TelSsInfo_t 282
TelSsRecord_t 283
TelSsUssdMsgInfo_t 284
TelTapiErrorInfo_t 285
TelTapiEvent_t 286
TelTapiTunnelInfo_t 287
TelTapiVersionInfo_t 288
TelUsimEccInfo_t 289
TelUSimServiceTable_t 290

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

- ComponentOverview.h (Component Programming Interface Overview Header Template for Doxygen) . . . 297
- ITapiCall.h (This file serves as a "C" header file defines functions for Tapi Call Services. It contains a sample set of function prototypes that would be required by applications) . 298
- ITapiNetText.h (This file serves as a "C" header file defines functions for Tapi NetText Services. It contains a sample set of function prototypes that would be required by applications) 300
- ITapiNetwork.h (This file serves as a "C" header file defines functions for Tapi Network. It contains a sample set of function prototypes that would be required by applications) . 302
- ITapiSim.h (This file serves as a "C" header file defines functions for Tapi SIM Services. It contains a sample set of function prototypes that would be required by applications) . 304
- ITapiSs.h (This file serves as a "C" header file defines functions for Tapi Supplementary Services. It contains a sample set of function prototypes that would be required by applications) 307
- TapiCommon.h (This file serves as a "C" header file defines functions for Tapi Common Interfaces like Callback Registration / De-registration. It contains a sample set of function prototypes that would be required by applications) . . . 309
- TelCall.h (This file serves as a "C" header file defines structures for Tapi Call Services. It contains a sample set of constants, enums, structs that would be required by applications) . . . 312
- TelNetText.h (This file serves as a "C" header file defines structures for Tapi Network Text Services. It contains a sample set of constants, enums,structs that would be required by applications) . . . 315
- TelNetwork.h (This file serves as a "C" header file defines structures for Tapi Network Services. It contains a sample set of constants, enums,structs that would be required by applications)317
- TelSim.h (This file serves as a "C" header file defines structures for Tapi SIM Services. It contains a sample set of constants, enums,structs that would be required by applications) 319
- TelSs.h (This file serves as a "C" header file defines structures for Tapi SS Services. It contains a sample set of constants, enums,structs that would be required by applications) 324
- TelUtility.h (This file serves as a "C" header file defines structures for Utility Services. It contains a sample set of constants, enums,structs that would be required by applications) 327

Chapter 5

Module Documentation

5.1 CallServices

Files

- file ITapiCall.h

  This file serves as a "C" header file defines functions for Tapi Call Services.
  It contains a sample set of function prototypes that would be required by applications.

Modules

- TelTapiCallSetup
- TelTapiCallAnswer
- TelTapiCallRelease
- TelTapiCallReleaseAll
- TelTapiCallSendDTMF
- TelTapiCallRetrieve
- TelTapiCallHold
- TelTapiCallJoin
- TelTapiCallSplit
- TelTapiCallExplicitTransfer
- TelTapiCallGetStatus
- TelTapiCallGetDuration
- TelTapiCallGetConferenceList

5.2 TelTapiCallSetup

Functions

- TapiResult_t TelTapiCallSetup (const TelCallSetupParams_t ∗pParams, unsigned int ∗pCallHandle, int ∗pRequestId)

  This function originates MO call setup. This is an asynchronous function. The function completes immediately and call setup proceeds in background.

5.2.1 Function Documentation

5.2.1.1 TapiResult_t TelTapiCallSetup (const TelCallSetupParams_t ∗ *pParams*, unsigned int ∗ *pCallHandle*, int ∗ *pRequestId*)

This function originates MO call setup. This is an asynchronous function. The function completes immediately and call setup proceeds in background.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- If call to TelTapiCallSetup() is successful, call handle and request ID are assigned by Telephony Server and returned to client by given Call Handle and Request Id parameters. MO call setup continues in background. There are multiple states involved in the call setup procedure. Upon request completion successful or not, various event notifications are sent to client along with data associated with event (Successful data or error information) based on the indication or response received at the Telephony Server.
- Setting up Emergency call
  - when TelCallType_t is voice_call, voice call is setup unless dial_number parameter is an emergency number known to telephony FW (like 112 for all 2G and above, other numbers may operator specific or device configuration specific, so Telephony FW will read relevant SIM files and refer to internal configuration records). If an emergency number is passed and TelCallType_t is voice_call, Telephony FW shall setup emergency phone call.
  - When TelCallType_t is emergency call, emergency call is always set up.

Warning:

- None.

Parameters:

← ***pParams*** • TelCallSetupParams_t contains the necessary parameters like supplementary information, the type of the call (whether it is a voice call or data call etc), identity mode option (Identity to be shown or hidden etc) and the Closed User Group (CUG) information for setting up the call.

→ ***pCallHandle*** • Unique handle for referring the call

→ ***pRequestId*** • Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

TAPI_EVENT_CALL_SETUP_CNF is sent to the application when the lower layers get the call originate response. Asynchronous return status is indicated by TelCallCause_t and call handle is sent in the event data.

Pre Conditions:

- None.

Post Conditions:

- None.

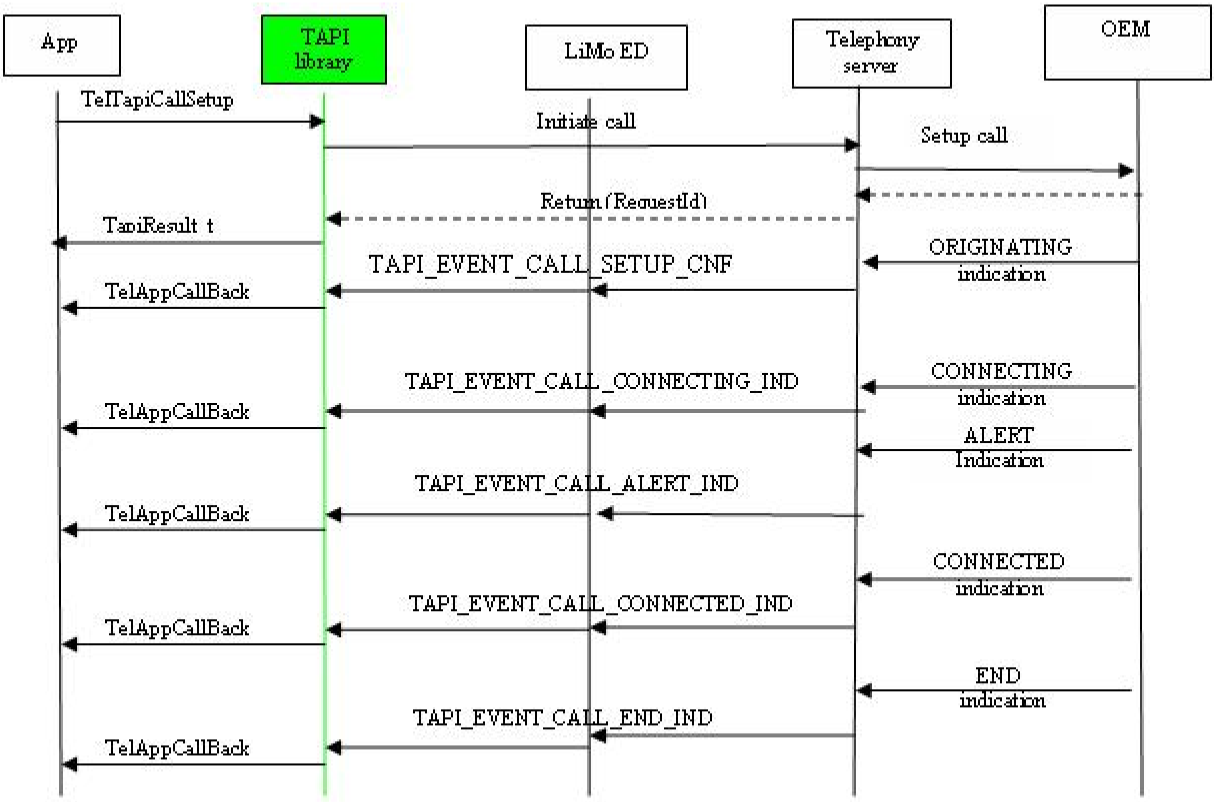
Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.



5.3 TelTapiCallAnswer

Functions

- TapiResult_t TelTapiCallAnswer (unsigned int CallHandle, TelCallAnswerType_t AnsType, int ∗pRequestId)

  This function is used to answer the incoming call by accepting or rejecting the call.

5.3.1 Function Documentation

5.3.1.1 TapiResult_t TelTapiCallAnswer (unsigned int *CallHandle*, TelCallAnswerType_t *AnsType*, int ∗ *pRequestId*)

This function is used to answer the incoming call by accepting or rejecting the call.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***CallHandle***
- This is the unique handle for referring the call. This call handle is available to Application through incoming call (TAPI_EVENT_CALL_INCOM_IND) event.

← ***AnsType***
- Answer type (answer / reject)

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_ANSWER_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle is sent in the event data.
- TAPI_EVENT_CALL_CONNECTED_IND / TAPI_EVENT_CALL_END_IND event will be notified in case of call is accepted/rejected.

Pre Conditions:

- Call associated with call handle should be TAPI_CALL_STATE_INCOM state otherwise the API fails and there can be a maximum of 1 existing call.

Post Conditions:

- None.

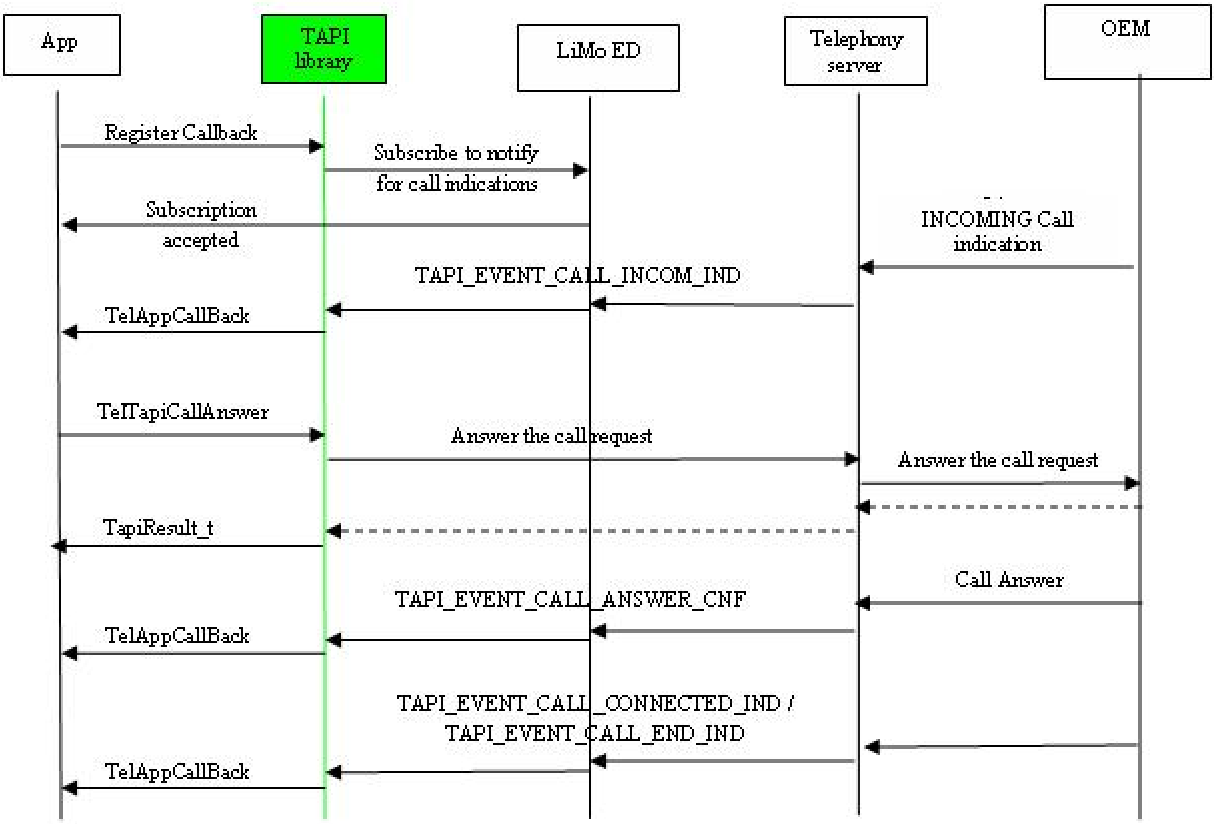
Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.



5.4 TelTapiCallRelease

Functions

- TapiResult_t TelTapiCallRelease (unsigned int CallHandle, int ∗pRequestId)
  This function releases the call identified by Call Handle irrespective of call is hold or active state.

5.4.1 Function Documentation

5.4.1.1 TapiResult_t TelTapiCallRelease (unsigned int *CallHandle*, int ∗ *pRequestId*)

This function releases the call identified by Call Handle irrespective of call is hold or active state.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- In case of MPTY Call handle, Call end event is returned for each call in MPTY.

Warning:

- None.

Parameters:

← ***CallHandle***	• Unique handle for referring the call.
→ ***pRequestId***	• Unique identifier for a particular request. • Request Id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_END_IND and the associated data is TelCallEndInfo_t. Asynchronous return status is indicated by TelCallCause_t.

Pre Conditions:

- Call handle should be valid and there should be an existing call in Active/hold state.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.5 TelTapiCallReleaseAll

Functions

- TapiResult_t TelTapiCallReleaseAll (int ∗pRequestId)

  This function releases the all the calls irrespective of the call is hold or active state. Due to this API request, lower layers will release the call and END Indication Event will be sent to all the Applications.

5.5.1 Function Documentation

5.5.1.1 TapiResult_t TelTapiCallReleaseAll (int ∗ *pRequestId*)

This function releases the all the calls irrespective of the call is hold or active state. Due to this API request, lower layers will release the call and END Indication Event will be sent to all the Applications.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- This asynchronous event is received for each call getting released.The event associated with this request is TAPI_EVENT_CALL_END_IND and the associated data is TelCallEndInfo_t. There will be no request id present associated with this TAPI_EVENT_CALL_END_IND when this API is called. Once all the calls are released, Telephony server will indicate application through TAPI Library by sending TAPI_EVENT_CALL_RELEASE_ALL_CNF. Asynchronous return status is indicated by TelCallCause_t and there is no associated event data.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.6 TelTapiCallSendDTMF

Functions

- TapiResult_t TelTapiCallSendDTMF (const char ∗pDtmfString, int ∗pRequestId)

  This function sends one or more DTMF digits during the call.

5.6.1 Function Documentation

5.6.1.1 TapiResult_t TelTapiCallSendDTMF (const char ∗ *pDtmfString*, int ∗ *pRequestId*)

This function sends one or more DTMF digits during the call.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

-There will be a single asynchronous notification for all the dtmf digits sent. If the users of this API need an asynchronous response for each DTMF digit then User has to call this API multiple times passing each single dtmf digit in pDtmfString.

Warning:

- None.

Parameters:

← ***pDtmfString***	• A Null terminated DTMF string to be sent from MS.
→ ***pRequestId***	• Unique identifier for a particular request. • Request Id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_SEND_DTMF_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle is sent in the event data.

Pre Conditions:

- Active call should be present.

Post Conditions:

- None.

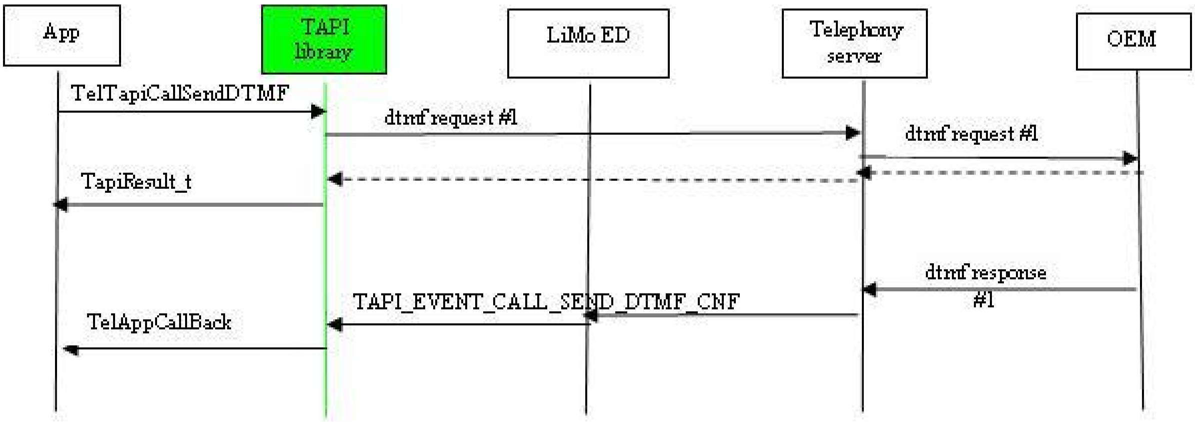
Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.



5.7 TelTapiCallRetrieve

Functions

- TapiResult_t TelTapiCallRetrieve (unsigned int CallHandle, int ∗pRequestId)
  This function retrieves the held call. The call identified by Call Handle must be in held state.

5.7.1 Function Documentation

5.7.1.1 TapiResult_t TelTapiCallRetrieve (unsigned int *CallHandle*, int ∗ *pRequestId*)

This function retrieves the held call. The call identified by Call Handle must be in held state.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Call duration for the call will be calculated from the moment call has been connected until call is released.

Warning:

- None.

Parameters:

← ***CallHandle*** • Unique handle for referring the call.

→ ***pRequestId*** • Unique identifier for a particular request.

- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_RETRIEVE_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle is sent in the event data.

Pre Conditions:

- Call should be in held state in order to retrieve into active state and no active call present.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.8 TelTapiCallHold

Functions

- TapiResult_t TelTapiCallHold (unsigned int CallHandle, int ∗pRequestId)
  This function puts the given call on hold. The call identified by Call Handle should be in active state.

5.8.1 Function Documentation

5.8.1.1 TapiResult_t TelTapiCallHold (unsigned int *CallHandle*, int ∗ *pRequestId*)

This function puts the given call on hold. The call identified by Call Handle should be in active state.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Call duration for the call will be calculated from the time call has been connected until call is released, i.e. Time duration during the call on hold will also be counted.

Warning:

- None.

Parameters:

← ***CallHandle*** • Unique handle for referring the call.

→ ***pRequestId*** • Unique identifier for a particular request.

- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_HOLD_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle is sent in the event data.

Pre Conditions:

- The call identified by the Call Handle should be in active state.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.9 TelTapiCallJoin

Functions

- TapiResult_t TelTapiCallJoin (unsigned int CallHandle1, unsigned int CallHandle2, int ∗pRequestId)

  This function joins the given two calls (one call in active conversation state and other call in held state) into conference.

5.9.1 Function Documentation

5.9.1.1 TapiResult_t TelTapiCallJoin (unsigned int *CallHandle1*, unsigned int *CallHandle2*, int ∗ *pRequestId*)

This function joins the given two calls (one call in active conversation state and other call in held state) into conference.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Call handle of an active call or held call will be made as MPTY-id, and the other call handle will join in a conference).

Warning:

- None.

Parameters:

← ***CallHandle1***	• The unique handle can be either an active call or held call.
← ***CallHandle2***	• Unique call handle.
→ ***pRequestId***	• Unique identifier for a particular request. • Request Id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_SETUPCONFERENCE_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle of the MPTY call is sent in the event data.

Pre Conditions:

- For a Multiparty call or for joining two calls into conference, there should be one call in active state and another call in held state.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.10 TelTapiCallSplit

Functions

- TapiResult_t TelTapiCallSplit (unsigned int CallHandle, int ∗pRequestId)

  This function triggers to splits a private call from multiparty call. This API allows creating a private communication with one of remote parties in a multiparty session. TAPI client application has to pass the call handle which needs to split from the multiparty call.

5.10.1 Function Documentation

5.10.1.1 TapiResult_t TelTapiCallSplit (unsigned int *CallHandle*, int ∗ *pRequestId*)

This function triggers to splits a private call from multiparty call. This API allows creating a private communication with one of remote parties in a multiparty session. TAPI client application has to pass the call handle which needs to split from the multiparty call.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- The list of call handles present in the conference call is retrieved using TelTapiCallGetConferenceList API

Warning:

- None.

Parameters:

← ***CallHandle***
- handle of call to be made private. Call handle referring to the call that is to be split from the conference (call to be made private).

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_SPLITCONFERENCE_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle for the split call is sent in the event data.

Pre Conditions:

- Call should be in multiparty conference call.

Post Conditions:

- Split call will be the active call and the conference will be held call.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.11 TelTapiCallExplicitTransfer

Functions

- TapiResult_t TelTapiCallExplicitTransfer (unsigned int CallHandle, int ∗pRequestId)
  This function triggers to do an explicit call transfer by connecting the two parties where in one party being active (active state) and another party being held (held state).

5.11.1 Function Documentation

5.11.1.1 TapiResult_t TelTapiCallExplicitTransfer (unsigned int *CallHandle*, int ∗ *pRequestId*)

This function triggers to do an explicit call transfer by connecting the two parties where in one party being active (active state) and another party being held (held state).

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***CallHandle***	• Call handle of an active call.
→ ***pRequestId***	• Unique identifier for a particular request.

- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

- The event associated with this request is TAPI_EVENT_CALL_TRANSFER_CNF. Asynchronous return status is indicated by TelCallCause_t and call handle is sent in the event data. Call handle in event data is active call handle which is used in the TelTapiCallExplicitTransfer request.

Pre Conditions:

- In order to call transfer, Served mobile subscriber should have 2 calls in one in active state and another one Call in Held state.

Post Conditions:

- When the request has been completed successfully, Call end indication will be sent to both the calls (active and held).

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.12 TelTapiCallGetStatus

Functions

- TapiResult_t TelTapiCallGetStatus (unsigned int CallHandle, TelCallStatus_t *pCallStatus)
  This function gets status for the current call identified by Call Handle whenever application wants the call status. Call handle must be valid.

5.12.1 Function Documentation

5.12.1.1 TapiResult_t TelTapiCallGetStatus (unsigned int *CallHandle*, TelCallStatus_t * *pCallStatus*)

This function gets status for the current call identified by Call Handle whenever application wants the call status. Call handle must be valid.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***CallHandle***	• Unique handle for referring the call.
→ ***pCallStatus***	• Call status information like destination number, call direction (MO or MT), call type (voice or data etc), whether the call is in conference state or not, present call state etc are returned through this parameter.

Async Response Message:

- None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.13 TelTapiCallGetDuration

Functions

- TapiResult_t TelTapiCallGetDuration (unsigned int CallHandle, unsigned int ∗pDurationInSecs)
  This function gets duration of the given call. This is a synchronous function. Duration is accounted from the moment the connection is established, i.e. call goes into active state for first time.

5.13.1 Function Documentation

5.13.1.1 TapiResult_t TelTapiCallGetDuration (unsigned int *CallHandle*, unsigned int ∗ *pDurationInSecs*)

This function gets duration of the given call. This is a synchronous function. Duration is accounted from the moment the connection is established, i.e. call goes into active state for first time.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***CallHandle*** • Unique handle for referring the call.

→ ***pDurationInSecs*** • The total call duration in seconds from the call connecting time to the present time is returned through this parameter.

Async Response Message:

- None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.14 TelTapiCallGetConferenceList

Functions

- TapiResult_t TelTapiCallGetConferenceList (unsigned int CallHandle, unsigned int ∗pCallList, int ∗pNoOfCalls)

  This is a synchronous function returns all call handles within the given conference call.

5.14.1 Function Documentation

5.14.1.1 TapiResult_t TelTapiCallGetConferenceList (unsigned int *CallHandle*, unsigned int ∗ *pCallList*, int ∗ *pNoOfCalls*)

This is a synchronous function returns all call handles within the given conference call.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- Call should be a Multi-party conference call in order to return actual no of calls and call handles in the conference calls.

Warning:

- None.

Parameters:

← ***CallHandle*** • Handle of call which is associated with the conference.

→ ***pCallList*** • list of call joined in the conference call. In case there is no active conference. The list will be zero and number of calls parameter value will also be zero. Maximum number of calls in a conference can be upto 5 (Based on 3GPP TS 22.084). Memory allocation for calllist is integer array of size 5.

→ ***pNoOfCalls*** • Number of the calls present in conference.

Async Response Message:

-None.

Pre Conditions:

-None.

Post Conditions:

-None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.15 NetTextServices

Files

- file ITapiNetText.h

  This file serves as a "C" header file defines functions for Tapi NetText Services.
  It contains a sample set of function prototypes that would be required by applications.

Modules

- TelTapiSmsSend
- TelTapiSmsRead
- TelTapiSmsWrite
- TelTapiSmsDelete
- TelTapiSmsGetCount
- TelTapiSmsCBSetConfig
- TelTapiSmsCBGetConfig
- TelTapiSmsSetSCA
- TelTapiSmsGetSCA
- TelTapiSmsSetPreferredBearer
- TelTapiSmsGetPreferredBearer
- TelTapiSmsGetParameters
- TelTapiSmsSetParameters
- TelTapiSmsSendDeliveryReport

5.16 TelTapiSmsSend

Functions

- TapiResult_t TelTapiSmsSend (const TelSmsDatapackageInfo_t ∗pDataPackage, BOOL bMoreMsgToSend, int ∗pRequestId)

  This API is used to send an SMS message to the network. This API interface allows transmitting SMS PDU as defined by 3GPP TS 23.040 (SMS-SUBMIT, SMS-COMMAND) to lower layers. This API allows keeping the dedicated link at lower layers by passing information as more messages to send in parameters. This will enable not to release if a dedicated connection is used for transmission.

5.16.1 Function Documentation

5.16.1.1 TapiResult_t TelTapiSmsSend (const TelSmsDatapackageInfo_t ∗ *pDataPackage*, BOOL *bMoreMsgToSend*, int ∗ *pRequestId*)

This API is used to send an SMS message to the network. This API interface allows transmitting SMS PDU as defined by 3GPP TS 23.040 (SMS-SUBMIT, SMS-COMMAND) to lower layers. This API allows keeping the dedicated link at lower layers by passing information as more messages to send in parameters. This will enable not to release if a dedicated connection is used for transmission.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pDataPackage***
- Contains SMS-SUBMIT TPDU

← ***bMoreMsgToSend***
- will be set to TRUE when there are more than one message to be send or Multiple segmented concatenated message to be send, otherwise FALSE. This flag indicates OEM that more messages to send.

→ ***pRequestId***
- Unique identifier for a particular request.
- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

SMS sent status is returned in this event TAPI_EVENT_NETTEXT_SENTSTATUS_CNF and the event data associated with this event is TelSmsDatapackageInfo_t. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- SMS Preferred bearer has to be set using TelTapiSmsSetPreferredBearer API if there is any preference from User. Otherwise default preferred bearer (This will be set by Vendor based on operator preference/support) will be used.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.17 TelTapiSmsRead

Functions

- TapiResult_t TelTapiSmsRead (int Index, int ∗pRequestId)

  This function reads a message from SIM card storage. SIM index (index used when storing the SMS) number to be passed to in order to read particular SMS message. Total number of messages can be retrieved by TelTapiSmsGetCount function.

5.17.1 Function Documentation

5.17.1.1 TapiResult_t TelTapiSmsRead (int *Index*, int ∗ *pRequestId*)

This function reads a message from SIM card storage. SIM index (index used when storing the SMS) number to be passed to in order to read particular SMS message. Total number of messages can be retrieved by TelTapiSmsGetCount function.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***Index***	• Index number of the message to be read
→ ***pRequestId***	• Unique identifier for a particular request. • pRequestId value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_READ_SMS_CNF and the event data is TelSmsData_t. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.18 TelTapiSmsWrite

Functions

- TapiResult_t TelTapiSmsWrite (TelSmsMsgStatus_t MsgStatus, const TelSmsDatapackageInfo_t ∗pDataPackage, int ∗pRequestId)

  This function writes a message to SIM storage area.

5.18.1 Function Documentation

5.18.1.1 TapiResult_t TelTapiSmsWrite (TelSmsMsgStatus_t *MsgStatus*, const TelSmsDatapackageInfo_t ∗ *pDataPackage*, int ∗ *pRequestId*)

This function writes a message to SIM storage area.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***MsgStatus*** • Message status information

← ***pDataPackage*** • SMS message structure.

→ ***pRequestId*** • Unique identifier for a particular request.

- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

TAPI library will send TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF asynchronous event for this request when received from TS. The associated data is Sim Index in which the message stored. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.19 TelTapiSmsDelete

Functions

- TapiResult_t TelTapiSmsDelete (signed short Index, int ∗pRequestId)
  This function deletes a message specified by index (SIM index used when storing the sms) number from the SIM storage area.

5.19.1 Function Documentation

5.19.1.1 TapiResult_t TelTapiSmsDelete (signed short *Index*, int ∗ *pRequestId*)

This function deletes a message specified by index (SIM index used when storing the sms) number from the SIM storage area.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***Index*** • Message index to delete.

→ ***pRequestId*** • Unique identifier for a particular request.

- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The asynchronous event associated is TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF and the event data is Sim index from which the message has been deleted. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t).

Prospective Clients:

External Apps.

5.20 TelTapiSmsGetCount

Functions

- TapiResult_t TelTapiSmsGetCount (int ∗pRequestId)

  This function is used to retrieve message count information like Total record count, used record count and the indices list for SIM Storage. This is an asynchronous function and the response will be sent back by event notification.

5.20.1 Function Documentation

5.20.1.1 TapiResult_t TelTapiSmsGetCount (int ∗ *pRequestId*)

This function is used to retrieve message count information like Total record count, used record count and the indices list for SIM Storage. This is an asynchronous function and the response will be sent back by event notification.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_GET_COUNT_CNF and the event data is TelSmsStoredMsgCountInfo_t. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

- External Apps.

5.21 TelTapiSmsCBSetConfig

Functions

- TapiResult_t TelTapiSmsCBSetConfig (const TelSmsCbConfig_t ∗pCBConfig, int ∗pRequestId)
  This function is used to set SMS CB Message Identifiers in the appropriate EF-CBMI/EF-CBMIR file in (U)SIM. This function also indicates underlying provider to enable or disable CB Channel on which CB messages are received.

5.21.1 Function Documentation

5.21.1.1 TapiResult_t TelTapiSmsCBSetConfig (const TelSmsCbConfig_t ∗ *pCBConfig*, int ∗ *pRequestId*)

This function is used to set SMS CB Message Identifiers in the appropriate EF-CBMI/EF-CBMIR file in (U)SIM. This function also indicates underlying provider to enable or disable CB Channel on which CB messages are received.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pCBConfig*** • This structure contains parameters to set SMS CB configuration
(i) Cell broadcast service state. (Enabled/disabled.)
(ii) Cell broadcast message identifier, which denotes whether all cell broadcast message identifiers are selected or only some of those.
(iii) The list count of message identifiers to be set.
(iv) The actual list of message identifiers to be set. Each identifier occupies 2 bytes. And the list ranges from 0x00 to 0xff.
These message identifiers denote the source and type of the cb message. Based on the message identifiers set in the SIM, the ME receives CB messages with selected message identifiers.

→ ***pRequestId*** • Unique identifier for a particular request.

- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_SET_REQUEST_CNF and there is no data associated with this event. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.22 TelTapiSmsCBGetConfig

Functions

- TapiResult_t TelTapiSmsCBGetConfig (int ∗pRequestId)
  This function is used to retrieve SMS CB configuration options from EFcbmi file in the SIM.

5.22.1 Function Documentation

5.22.1.1 TapiResult_t TelTapiSmsCBGetConfig (int ∗ *pRequestId*)

This function is used to retrieve SMS CB configuration options from EFcbmi file in the SIM.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF and the event data is TelSmsCbConfig_t. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

- External Apps.

5.23 TelTapiSmsSetSCA

Functions

- TapiResult_t TelTapiSmsSetSCA (const TelSmsAddressInfo_t ∗pSCA, int ∗pRequestId)
  This function is used to set SMS Service Centre Address information in order to send the SMS.

5.23.1 Function Documentation

5.23.1.1 TapiResult_t TelTapiSmsSetSCA (const TelSmsAddressInfo_t ∗ *pSCA*, int ∗ *pRequestId*)

This function is used to set SMS Service Centre Address information in order to send the SMS.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- This API updates/sets service center address in Efsmsp file in the SIM storage.

Warning:

- None.

Parameters:

← ***pSCA*** • Contains the service centre address informations.

→ ***pRequestId*** • Unique identifier for a particular request.
  - pRequestId value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_SET_REQUEST_CNF and there is no data associated with this event. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t).

Prospective Clients:

External Apps.

5.24 TelTapiSmsGetSCA

Functions

- TapiResult_t TelTapiSmsGetSCA (TelSmsAddressInfo_t ∗pSCA)
  This function is used to get current SMS Service Centre Address information.

5.24.1 Function Documentation

5.24.1.1 TapiResult_t TelTapiSmsGetSCA (TelSmsAddressInfo_t ∗ *pSCA*)

This function is used to get current SMS Service Centre Address information.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pSCA***	• Contains the service centre address information received from the modem.

Async Response Message:

None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.25 TelTapiSmsSetPreferredBearer

Functions

- TapiResult_t TelTapiSmsSetPreferredBearer (TelSmsBearerType_t BearerType, int ∗pRequestId)
  This function is used to set SMS preferred bearer on which SMS has to be transmitted.

5.25.1 Function Documentation

5.25.1.1 TapiResult_t TelTapiSmsSetPreferredBearer (TelSmsBearerType_t *BearerType*, int ∗ *pRequestId*)

This function is used to set SMS preferred bearer on which SMS has to be transmitted.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***BearerType***	• Specifies the bearer type.
→ ***pRequestId***	• Unique identifier for a particular request.

- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_SET_REQUEST_CNF and there is no data associated with this event. Asynchronous return status is indicated by TelSmsResponse_t

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.26 TelTapiSmsGetPreferredBearer

Functions

- TapiResult_t TelTapiSmsGetPreferredBearer (int ∗pRequestId)
  This function is used to get SMS preferred bearer on which SMS has to be transmitted. This is an asynchronous function.

5.26.1 Function Documentation

5.26.1.1 TapiResult_t TelTapiSmsGetPreferredBearer (int ∗ *pRequestId*)

This function is used to get SMS preferred bearer on which SMS has to be transmitted. This is an asynchronous function.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId*** • Unique identifier for a particular request.
- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF and the associated event data is TelSmsBearerType_t. Asynchronous return status is indicated by TelSmsResponse_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.27 TelTapiSmsGetParameters

Functions

- TapiResult_t TelTapiSmsGetParameters (const int SmsRecordIndex, int ∗pRequestId)
  This function is used to get the SMS parameters for a particular SMS (sent/received) based on the SIM index where it is stored. This is an asynchronous function.

5.27.1 Function Documentation

5.27.1.1 TapiResult_t TelTapiSmsGetParameters (const int *SmsRecordIndex*, int ∗ *pRequestId*)

This function is used to get the SMS parameters for a particular SMS (sent/received) based on the SIM index where it is stored. This is an asynchronous function.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***SmsRecordIndex*** • Sim index where the SMS is stored

→ ***pRequestId*** • Unique identifier for a particular request.

  - pRequest Id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_GET_PARAM_CNF and the event data is TelSmsParams_t. Asynchronous return status is indicated by TelSmsResponse_t

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.28 TelTapiSmsSetParameters

Functions

- TapiResult_t TelTapiSmsSetParameters (const TelSmsParams_t SmsSetParameters, int ∗pRequestId)

  This function is used to set the short message service header parameters which are used in the origination of MO messages.

5.28.1 Function Documentation

5.28.1.1 TapiResult_t TelTapiSmsSetParameters (const TelSmsParams_t *SmsSetParameters*, int ∗ *pRequestId*)

This function is used to set the short message service header parameters which are used in the origination of MO messages.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- This API updates/sets SMS Parameters in Efsmsp file in the SIM storage

Warning:

- None.

Parameters:

← ***SmsSetParameters***
- Contains SMS header Parameter information like service center address, destination address, validity period, protocol identifier, data coding scheme.

→ ***pRequestId***
- Unique identifier for a particular request.
  - pRequestId value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_SET_REQUEST_CNF and there is no data associated with this event. Asynchronous return status is indicated by TelSmsResponse_t

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.29 TelTapiSmsSendDeliveryReport

Functions

- TapiResult_t TelTapiSmsSendDeliveryReport (const TelSmsDatapackageInfo_t ∗pDataPackage, int ∗pRequestId)

  This function is used to send a delivery report for the received incoming SMS to the network.

5.29.1 Function Documentation

5.29.1.1 TapiResult_t TelTapiSmsSendDeliveryReport (const TelSmsDatapackageInfo_t ∗ *pDataPackage*, int ∗ *pRequestId*)

This function is used to send a delivery report for the received incoming SMS to the network.

Sync (or) Async:

This is a Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pDataPackage*** • Contains SMS Data package..

→ ***pRequestId*** • Unique identifier for a particular request.

- pRequestId value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF and there is no associated event data. Asynchronous return status is indicated by TelSmsResponse_t

Pre Conditions:

- SMS-DELIVER REPORT message structure should be in TPDU format as specified by 3GPP TS 23.040 SMS TPDU.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.30 NetworkServices

Files

- file ITapiNetwork.h

  This file serves as a "C" header file defines functions for Tapi Network.
  It contains a sample set of function prototypes that would be required by applications.

Modules

- TelTapiNwSetConfiguration
- TelTapiNwSelectNetwork
- TelTapiNwCancelManualSearch
- TelTapiNwCancelManualSelection
- TelTapiNwGetNetworkInfo
- TelTapiNwSearchNetwork
- TelTapiNwSetSelectionMode
- TelTapiNwGetSelectionMode
- TelTapiNwSetServiceDomain
- TelTapiNwSetNetworkOrder
- TelTapiNwSetNetworkMode
- TelTapiNwSetNetworkBand

5.31 TelTapiNwSetConfiguration

Functions

- TapiResult_t TelTapiNwSetConfiguration (const TelNetworkConfigSettings_t ∗pConfigNetworkInfo, int ∗pRequestId)

  This API allows to set the network configuration settings at OEM based on which network registration will happen.

5.31.1 Function Documentation

5.31.1.1 TapiResult_t TelTapiNwSetConfiguration (const TelNetworkConfigSettings_t ∗ *pConfigNetworkInfo*, int ∗ *pRequestId*)

This API allows to set the network configuration settings at OEM based on which network registration will happen.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pConfigNetworkInfo***
- Contains Network information like network order, mode, service domain, etc.

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_CONFIG_CNF and there is no event data associated with this event and the asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.32 TelTapiNwSelectNetwork

Functions

- TapiResult_t TelTapiNwSelectNetwork (const TelNetworkSystemId_t ∗pSystemId, int ∗pRequestId)

  This function requests the lower layers to select the network (PLMN) which has been selected by the user from the Network List displayed to the User.

5.32.1 Function Documentation

5.32.1.1 TapiResult_t TelTapiNwSelectNetwork (const TelNetworkSystemId_t ∗ *pSystemId*, int ∗ *pRequestId*)

This function requests the lower layers to select the network (PLMN) which has been selected by the user from the Network List displayed to the User.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- This function passes the TelNetworkSystemId_t information which is received in Search Network Confirmation event.

Warning:

- None.

Parameters:

← ***pSystemId***
- System Id consists systemid format, system type, plmnid and information on whether mnc includes pcs digit to identify whether it belongs to 310-xxx plmns or not.

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SELECT_CNF and the event data is TelNetworkInfo_t. Asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.33 TelTapiNwCancelManualSearch

Functions

- TapiResult_t TelTapiNwCancelManualSearch (int ∗pRequestId)
  This function is called to cancel the triggered manual network search.

5.33.1 Function Documentation

5.33.1.1 TapiResult_t TelTapiNwCancelManualSearch (int ∗ *pRequestId*)

This function is called to cancel the triggered manual network search.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- Manual network search is already triggered.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.34 TelTapiNwCancelManualSelection

Functions

- TapiResult_t TelTapiNwCancelManualSelection (int ∗pRequestId)

  This function is called to cancel manual network selection where user has already selected a network from the network list.

5.34.1 Function Documentation

5.34.1.1 TapiResult_t TelTapiNwCancelManualSelection (int ∗ *pRequestId*)

This function is called to cancel manual network selection where user has already selected a network from the network list.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId*** • Unique identifier for a particular request.

- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- User should have selected a network from available network list.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.35 TelTapiNwGetNetworkInfo

Functions

- TapiResult_t TelTapiNwGetNetworkInfo (int ∗pRequestId)
  This function provides an option to get the current network registration status information.

5.35.1 Function Documentation

5.35.1.1 TapiResult_t TelTapiNwGetNetworkInfo (int ∗ *pRequestId*)

This function provides an option to get the current network registration status information.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF and the event data is TelNetworkInfo_t. Asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.36 TelTapiNwSearchNetwork

Functions

- TapiResult_t TelTapiNwSearchNetwork (int ∗pRequestId)
  This function sends a request to do manual network selection to search for the available networks and provide the Network List to the User/Application.

5.36.1 Function Documentation

5.36.1.1 TapiResult_t TelTapiNwSearchNetwork (int ∗ *pRequestId*)

This function sends a request to do manual network selection to search for the available networks and provide the Network List to the User/Application.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId*** • Unique identifier for a particular request.
  - Request Id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SEARCH_CNF and the event data is TelNetworkList_t. Asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- TelTapiNwSetSelectionMode API is called prior to this API with TelNetworkSelectionMode_t as TAPI_NETWORK_SELECTIONMODE_MANUAL.

Post Conditions:

- In the available network list, user can select one of the networks successfully.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.37 TelTapiNwSetSelectionMode

Functions

- TapiResult_t TelTapiNwSetSelectionMode (TelNetworkSelectionMode_t NwSelectionMode, int ∗pRequestId)

  This function is called when user does Automatic or Manual Network Selection. Request for Automatic NW selection or Manual NW selection will allow OEM provider to do PLMN Selection based on network selection preference.

5.37.1 Function Documentation

5.37.1.1 TapiResult_t TelTapiNwSetSelectionMode (TelNetworkSelectionMode_t *NwSelectionMode*, int ∗ *pRequestId*)

This function is called when user does Automatic or Manual Network Selection. Request for Automatic NW selection or Manual NW selection will allow OEM provider to do PLMN Selection based on network selection preference.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- If this API is called with Manual Network Selection, then Application shall call TelTapiNwSearchNetwork API to trigger Manual selection so that network list can be provided to the User.

Warning:

- None.

Parameters:

← ***NwSelectionMode***	• Mode of selection i.e. Automatic or Manual
→ ***pRequestId***	• Unique identifier for a particular request. • Request Id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.38 TelTapiNwGetSelectionMode

Functions

- TapiResult_t TelTapiNwGetSelectionMode (int ∗pRequestId)

  This function requests for the present network selection mode i.e. automatic or manual.

5.38.1 Function Documentation

5.38.1.1 TapiResult_t TelTapiNwGetSelectionMode (int ∗ *pRequestId*)

This function requests for the present network selection mode i.e. automatic or manual.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

pRequestId
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF and the event data is TelNetworkSelectionMode_t. Asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.39 TelTapiNwSetServiceDomain

Functions

- TapiResult_t TelTapiNwSetServiceDomain (TelNetworkServiceDomain_t ServiceDomain, int ∗pRequestId)

  This function is called when User/application wants to configure the service domain to only CS or Only PS or Both. This API triggers the underlying protocol stack to do register with Network for only CS services or only PS services or both based on the option set using this API.

5.39.1 Function Documentation

5.39.1.1 TapiResult_t TelTapiNwSetServiceDomain (TelNetworkServiceDomain_t *ServiceDomain*, int ∗ *pRequestId*)

This function is called when User/application wants to configure the service domain to only CS or Only PS or Both. This API triggers the underlying protocol stack to do register with Network for only CS services or only PS services or both based on the option set using this API.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

ServiceDomain	• Specifies the type of Service domain (Packet switch, circuit switch,etc)
pRequestId	• Unique identifier for a particular request. • Request Id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SET_SVC_DOMAIN_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.40 TelTapiNwSetNetworkOrder

Functions

- TapiResult_t TelTapiNwSetNetworkOrder (TelNetworkMode_t ∗pPrefNwOrder, int ∗pRequestId)

  This API allows lower layers to scan based on preferred network order based options passed.

5.40.1 Function Documentation

5.40.1.1 TapiResult_t TelTapiNwSetNetworkOrder (TelNetworkMode_t ∗ *pPrefNwOrder*, int ∗ *pRequestId*)

This API allows lower layers to scan based on preferred network order based options passed.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- If network order list array holds the value "TAPI_NETWORK_MODE_AUTOMATIC" then other following values in the list shall be ignored.

Warning:

- None.

Parameters:

← ***pPrefNwOrder***
- This parameter is an array of size TAPI_NW_ORDER_MAX(which is defined as 5) of TelNetworkMode_t type specifies the preferred network order for camping.

→ ***pRequestId***
- Unique identifier for a particular request.
- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SETNWORDER_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.41 TelTapiNwSetNetworkMode

Functions

- TapiResult_t TelTapiNwSetNetworkMode (TelNetworkMode_t NwMode, int ∗pRequestId)

  This function allows to configure the Network Mode (Automatic, GSM, WCDMA) at OEM provider based on which lower layers can scan only those radio access technologies for registering with network.

5.41.1 Function Documentation

5.41.1.1 TapiResult_t TelTapiNwSetNetworkMode (TelNetworkMode_t *NwMode*, int ∗ *pRequestId*)

This function allows to configure the Network Mode (Automatic, GSM, WCDMA) at OEM provider based on which lower layers can scan only those radio access technologies for registering with network.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***NwMode***	• Specifies the network mode i.e. Automatic, WCDMA, GSM, CDMA and WLAN.
→ ***pRequestId***	• Unique identifier for a particular request. • Request Id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SETNWMODE_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.42 TelTapiNwSetNetworkBand

Functions

- TapiResult_t TelTapiNwSetNetworkBand (TelNetworkBandPreferred_t BandMode, TelNetworkBand_t Band, int ∗pRequestId)

  This function is invoked to set the network band and allows the underlying OEM provider to scan the set band.

5.42.1 Function Documentation

5.42.1.1 TapiResult_t TelTapiNwSetNetworkBand (TelNetworkBandPreferred_t *BandMode*, TelNetworkBand_t *Band*, int ∗ *pRequestId*)

This function is invoked to set the network band and allows the underlying OEM provider to scan the set band.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***BandMode*** • Band preference indicates the band provide to be preferred or select the band .

← ***Band*** • This enumeration defines different network Bands that user can choose.

→ ***pRequestId*** • Unique identifier for a particular request.

- Request Id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_NETWORK_SETNWBAND_CNF and there is no event data associated with this event and asynchronous return status is indicated by TelNetworkOperationCause_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.43 SIMServices

Files

- file ITapiSim.h

  This file serves as a "C" header file defines functions for Tapi SIM Services.
  It contains a sample set of function prototypes that would be required by applications.

Modules

- TelTapiSimGetFileInfo
- TelTapiSimReadFile
- TelTapiSimUpdateFile
- TelTapiSimVerifyPIN
- TelTapiSimChangePIN
- TelTapiSimDisablePIN
- TelTapiSimEnablePIN
- TelTapiSimUnblockPIN
- TelTapiSimDisableFDN
- TelTapiSimEnableFDN
- TelTapiSimGetDevCtxdata
- TelTapiSimGetIccIdInfo
- TelTapiSimGetPINStatus
- TelTapiSimGetPIN2Status
- TelTapiSimGetFDNStatus
- TelTapiSimGetSstInfo
- TelTapiSimGetUstInfo
- TelTapiSimGetImsiInfo
- TelTapiSimGetEccInfo
- TelTapiSimGetLanguagePrefInfo
- TelTapiSimGetLanguageIndInfo
- TelTapiSimGetCardType
- TelTapiSimPBGetStorageCount
- TelTapiSimPBGetEntryInfo
- TelTapiSimPBReadRecord
- TelTapiSimPBUpdateRecord
- TelTapiSimPBDeleteRecord

5.44 TelTapiSimGetFileInfo

Functions

- TapiResult_t TelTapiSimGetFileInfo (TelSimFileID_t FileId, int ∗pRequestId)

  This function is passed to OEM provider in order to fetch the Elementary File info. This functions fetches all the Information like Whether EF is present or not, If Present, no of records in EF and EF Data length etc. Application can make use this API to know the EF details prior calling SIM Read API.

5.44.1 Function Documentation

5.44.1.1 TapiResult_t TelTapiSimGetFileInfo (TelSimFileID_t *FileId*, int ∗ *pRequestId*)

This function is passed to OEM provider in order to fetch the Elementary File info. This functions fetches all the Information like Whether EF is present or not, If Present, no of records in EF and EF Data length etc. Application can make use this API to know the EF details prior calling SIM Read API.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None

Warning:

- None.

Parameters:

← ***FileId*** -TelSimFileID_t is the identifier for elementary files inside SIM.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_FILE_INFO_CNF and the event data is TelSimFileInfo_t. Asynchronous return status is indicated by TelSimAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.45 TelTapiSimReadFile

Functions

- TapiResult_t TelTapiSimReadFile (const TelSimReadFileRequestData_t ∗pReadFileReqData, int ∗pRequestId)

  This function is passed to OEM provider in order to read an elementary file.

5.45.1 Function Documentation

5.45.1.1 TapiResult_t TelTapiSimReadFile (const TelSimReadFileRequestData_t ∗ *pReadFileReqData*, int ∗ *pRequestId*)

This function is passed to OEM provider in order to read an elementary file.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None

Warning:

- None.

Parameters:

← ***pReadFileReqData*** -Different parameters like SIM file id to be read, SIM file type, offset for reading binary data, record index, the starting index, end index, length of data are passed as argument.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_READ_FILE_CNF and the event data is TelSimReadFileData_t. Asynchronous return status is indicated by TelSimAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.46 TelTapiSimUpdateFile

Functions

- TapiResult_t TelTapiSimUpdateFile (const TelSimUpdateFileRequestData_t ∗pUpdateFileReqData, int ∗pRequestId)

  This function is passed to OEM provider in-order to update records in an elementary File.

5.46.1 Function Documentation

5.46.1.1 TapiResult_t TelTapiSimUpdateFile (const TelSimUpdateFileRequestData_t ∗ *pUpdateFileReqData*, int ∗ *pRequestId*)

This function is passed to OEM provider in-order to update records in an elementary File.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None

Warning:

- None.

Parameters:

← ***pUpdateFileReqData*** -The information required to update elementary files in SIM is specified through this parameter.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_UPDATE_FILE_CNF and the event data is TelSimFileAccessGeneralResp_t. Asynchronous return status is indicated by TelSimAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.47 TelTapiSimVerifyPIN

Functions

- TapiResult_t TelTapiSimVerifyPIN (const TelSimPinData_t ∗pPinData, int ∗pRequestId)
  This function performs PIN verification. This function performs PIN verification based on the pin type passed along with pin data.

5.47.1 Function Documentation

5.47.1.1 TapiResult_t TelTapiSimVerifyPIN (const TelSimPinData_t ∗ *pPinData*, int ∗ *pRequestId*)

This function performs PIN verification. This function performs PIN verification based on the pin type passed along with pin data.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Based on Pin status (PIN/UPIN/PUK) is blocked, an unsolicited event TAPI_EVENT_SIM_-STATUS_IND is published to Applications

Warning:

- None.

Parameters:

← ***pPinData*** -PIN code, entered by the user.

→ ***pRequestId***
- Unique identifier for a particular request.
  - request_id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_VERIFY_PIN_CNF and the event data is TelSimPinInfo_t. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.48 TelTapiSimChangePIN

Functions

- TapiResult_t TelTapiSimChangePIN (TelSimPinType_t PinType, const char ∗pOldPinData, const char ∗pNewPinData, int ∗pRequestId)

  This function changes PIN code based on the pin type passed along with old pin data and new pin data.

5.48.1 Function Documentation

5.48.1.1 TapiResult_t TelTapiSimChangePIN (TelSimPinType_t *PinType*, const char ∗ *pOldPinData*, const char ∗ *pNewPinData*, int ∗ *pRequestId*)

This function changes PIN code based on the pin type passed along with old pin data and new pin data.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Based on Pin status (PIN/UPIN/PUK) is blocked, an unsolicited event TAPI_EVENT_SIM_STATUS_IND is published to Applications

Warning:

- None.

Parameters:

← ***PinType*** -Pin type information.

← ***pOldPinData*** -Old pin code, entered by the user. (Null terminated string)

← ***pNewPinData*** -New pin code, entered by the user. (Null terminated string)

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_CHANGE_PIN_CNF and the event data is TelSimPinInfo_t. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.49 TelTapiSimDisablePIN

Functions

- TapiResult_t TelTapiSimDisablePIN (const TelSimPinData_t ∗pPinData, int ∗pRequestId)
  This function disables the PIN based on the Pin type and pin data passed from Application.

5.49.1 Function Documentation

5.49.1.1 TapiResult_t TelTapiSimDisablePIN (const TelSimPinData_t ∗ *pPinData*, int ∗ *pRequestId*)

This function disables the PIN based on the Pin type and pin data passed from Application.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- For PIN2, disabling applies only for 3G USIMs and when PIN2 in NOT mapped to CHV2 of GSM application. Based on Pin status (PIN/UPIN/PUK) is blocked, an unsolicited event TAPI_EVENT_SIM_STATUS_IND is published to Applications.

Warning:

- None.

Parameters:

← ***pPinData*** -User entered Pin disable data and pin type

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_DISABLE_PIN_CNF and the event data is TelSimPinInfo_t. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.50 TelTapiSimEnablePIN

Functions

- TapiResult_t TelTapiSimEnablePIN (const TelSimPinData_t ∗pPinData, int ∗pRequestId)
  This function enables the PIN based on the Pin type and pin data passed from the Application.

5.50.1 Function Documentation

5.50.1.1 TapiResult_t TelTapiSimEnablePIN (const TelSimPinData_t ∗ *pPinData*, int ∗ *pRequestId*)

This function enables the PIN based on the Pin type and pin data passed from the Application.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- For PIN2, disabling applies only for 3G USIMs and when PIN2 in NOT mapped to CHV2 of GSM application. Based on Pin status (PIN/UPIN/PUK) is blocked, an unsolicited event TAPI_EVENT_SIM_STATUS_IND is published to Applications.

Warning:

- None.

Parameters:

← ***pPinData*** -User entered Pin disable data and pin type

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_ENABLE_PIN_CNF and the event data is TelSimPinInfo_t. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.51 TelTapiSimUnblockPIN

Functions

- TapiResult_t TelTapiSimUnblockPIN (const TelSimPinData_t ∗pPukPinData, const TelSimPinData_t ∗pNewPinData, int ∗pRequestId)

  This function performs unblock PIN operation based on PUK information passed along with unblock information entered by the user.

5.51.1 Function Documentation

5.51.1.1 TapiResult_t TelTapiSimUnblockPIN (const TelSimPinData_t ∗ *pPukPinData*, const TelSimPinData_t ∗ *pNewPinData*, int ∗ *pRequestId*)

This function performs unblock PIN operation based on PUK information passed along with unblock information entered by the user.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Based on Pin status (PIN/UPIN/PUK) is blocked, an unsolicited event TAPI_EVENT_SIM_STATUS_IND is published to Applications.

Warning:

- None.

Parameters:

← ***pPukPinData*** -Specifies unblocking PIN password

← ***pNewPinData*** -Specifies PIN password.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_UNBLOCK_PIN_CNF and the event data is TelSimPinInfo_t. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.52 TelTapiSimDisableFDN

Functions

- TapiResult_t TelTapiSimDisableFDN (const char ∗pPinData, int ∗pRequestId)
  This function disables FDN.

5.52.1 Function Documentation

5.52.1.1 TapiResult_t TelTapiSimDisableFDN (const char ∗ *pPinData*, int ∗ *pRequestId*)

This function disables FDN.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- An unsolicited event TAPI_EVENT_SIM_FDN_STATUS_IND is notified to indicate whenever FDN status is changed(disabled/enabled)

Warning:

- None.

Parameters:

← ***pPinData*** -User entered PIN2 data. (NULL terminated string)

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_DISABLE_FDN_CNF and there is no event data associated with this event. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.53 TelTapiSimEnableFDN

Functions

- TapiResult_t TelTapiSimEnableFDN (const char ∗pPinData, int ∗pRequestId)
  This function enables FDN.

5.53.1 Function Documentation

5.53.1.1 TapiResult_t TelTapiSimEnableFDN (const char ∗ *pPinData*, int ∗ *pRequestId*)

This function enables FDN.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- An unsolicited event TAPI_EVENT_SIM_FDN_STATUS_IND is notified to indicate whenever FDN status is changed(disabled/enabled)

Warning:

- None.

Parameters:

← ***pPinData*** -User entered PIN2 data. (NULL terminated string)

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_ENABLE_FDN_CNF and there is no event data associated with this event. Asynchronous return status is indicated by TelSimPinOperationResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.54 TelTapiSimGetDevCtxdata

Functions

- TapiResult_t TelTapiSimGetDevCtxdata (TelSimDeviceContext_t ∗pSimData)

  This function is used to get device context data of SIM. SIM device context data such as PIN Enabled status, FDN, BDN Status, IMSI information, ECC, LP, (U)SIM service table information, group identifier information will be passed.

5.54.1 Function Documentation

5.54.1.1 TapiResult_t TelTapiSimGetDevCtxdata (TelSimDeviceContext_t ∗ *pSimData*)

This function is used to get device context data of SIM. SIM device context data such as PIN Enabled status, FDN, BDN Status, IMSI information, ECC, LP, (U)SIM service table information, group identifier information will be passed.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimData*** TelSimDeviceContext_t contains the requested data, including sst/ust, IMSI, and other flags.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.55 TelTapiSimGetIccIdInfo

Functions

- TapiResult_t TelTapiSimGetIccIdInfo (TelSimIccIdInfo_t ∗pIccIdInfo)

  This function gets ICC id information.

5.55.1 Function Documentation

5.55.1.1 TapiResult_t TelTapiSimGetIccIdInfo (TelSimIccIdInfo_t ∗ *pIccIdInfo*)

This function gets ICC id information.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pIccIdInfo*** This parameter returns the ICC id information like icc length and unique card identification number.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.56 TelTapiSimGetPINStatus

Functions

- TapiResult_t TelTapiSimGetPINStatus (TelSimPinStatus_t ∗pPinStatus, TelSimPinType_t ∗pPinType)

  This function gets the current PIN (PIN1 or UPIN) status.

5.56.1 Function Documentation

5.56.1.1 TapiResult_t TelTapiSimGetPINStatus (TelSimPinStatus_t ∗ *pPinStatus*, TelSimPinType_t ∗ *pPinType*)

This function gets the current PIN (PIN1 or UPIN) status.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pPinStatus*** This parameter returns the current Pin status.

→ ***pPinType*** This parameter returns the current Pin (Pin1 or upin) type.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.57 TelTapiSimGetPIN2Status

Functions

- TapiResult_t TelTapiSimGetPIN2Status (TelSimPinStatus_t ∗pPin2Status)
  This function gets the current Pin2 status.

5.57.1 Function Documentation

5.57.1.1 TapiResult_t TelTapiSimGetPIN2Status (TelSimPinStatus_t ∗ *pPin2Status*)

This function gets the current Pin2 status.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pPin2Status*** This parameter returns the current Pin2 status.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.58 TelTapiSimGetFDNStatus

Functions

- TapiResult_t TelTapiSimGetFDNStatus (BOOL ∗pFdnStatus)
  This function is a synchronous API gets the current FDN status.

5.58.1 Function Documentation

5.58.1.1 TapiResult_t TelTapiSimGetFDNStatus (BOOL ∗ *pFdnStatus*)

This function is a synchronous API gets the current FDN status.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pFdnStatus*** This parameter returns the current fdn status whether enabled or disabled.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.59 TelTapiSimGetSstInfo

Functions

- TapiResult_t TelTapiSimGetSstInfo (TelSimSstInfo_t ∗pSimSstInfo)
  This function gets the SIM service table information.

5.59.1 Function Documentation

5.59.1.1 TapiResult_t TelTapiSimGetSstInfo (TelSimSstInfo_t ∗ *pSimSstInfo*)

This function gets the SIM service table information.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimSstInfo*** This parameter returns the SIM service table information like sst length, sst data etc.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.60 TelTapiSimGetUstInfo

Functions

- TapiResult_t TelTapiSimGetUstInfo (TelSimUstInfo_t ∗pSimUstInfo)
  This function gets the USIM service table information.

5.60.1 Function Documentation

5.60.1.1 TapiResult_t TelTapiSimGetUstInfo (TelSimUstInfo_t ∗ *pSimUstInfo*)

This function gets the USIM service table information.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimUstInfo*** This parameter returns the USIM service table information like sst length, sst data etc.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.61 TelTapiSimGetImsiInfo

Functions

- TapiResult_t TelTapiSimGetImsiInfo (TelSimImsiInfo_t ∗pSimImsiInfo)
  This function is a synchronous API gets the IMSI information.

5.61.1 Function Documentation

5.61.1.1 TapiResult_t TelTapiSimGetImsiInfo (TelSimImsiInfo_t ∗ *pSimImsiInfo*)

This function is a synchronous API gets the IMSI information.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimImsiInfo*** This parameter returns the IMSI information.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.62 TelTapiSimGetEccInfo

Functions

- TapiResult_t TelTapiSimGetEccInfo (TelSimEccInfo_t ∗pSimEccInfo, int ∗pCount)
  This function is a synchronous API used to get ECC data.

5.62.1 Function Documentation

5.62.1.1 TapiResult_t TelTapiSimGetEccInfo (TelSimEccInfo_t ∗ *pSimEccInfo*, int ∗ *pCount*)

This function is a synchronous API used to get ECC data.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimEccInfo*** This parameter returns the SIM emergency call code information like ecc length, service type etc.

→ ***pCount*** Specifies number of ECC records.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.63 TelTapiSimGetLanguagePrefInfo

Functions

- TapiResult_t TelTapiSimGetLanguagePrefInfo (TelSimLanguageInfo_t ∗pSimLpInfo)
  This API is used to get the language preference information.

5.63.1 Function Documentation

5.63.1.1 TapiResult_t TelTapiSimGetLanguagePrefInfo (TelSimLanguageInfo_t ∗ *pSimLpInfo*)

This API is used to get the language preference information.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimLpInfo*** This parameter returns the language preference information.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.64 TelTapiSimGetLanguageIndInfo

Functions

- TapiResult_t TelTapiSimGetLanguageIndInfo (TelSimLanguageInfo_t ∗pSimLiInfo)
  This API is used to get the language indication information.

5.64.1 Function Documentation

5.64.1.1 TapiResult_t TelTapiSimGetLanguageIndInfo (TelSimLanguageInfo_t ∗ *pSimLiInfo*)

This API is used to get the language indication information.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pSimLiInfo*** This parameter returns the language indication information.

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.65 TelTapiSimGetCardType

Functions

- TapiResult_t TelTapiSimGetCardType (TelSimCardType_t ∗pCardInfo)
  This function is a synchronous API which gets the Card Type whether SIM Card is a GSM SIM or a 3G SIM.

5.65.1 Function Documentation

5.65.1.1 TapiResult_t TelTapiSimGetCardType (TelSimCardType_t ∗ *pCardInfo*)

This function is a synchronous API which gets the Card Type whether SIM Card is a GSM SIM or a 3G SIM.

Sync (or) Async:

This is a synchronous function.

Important Notes:

- None

Warning:

- None.

Parameters:

→ ***pCardInfo*** This parameter returns the SIM card type information like whether it is GSM SIM card or USIM or it is of unknown type.

- TAPI_SIM_CARD_TYPE_UNKNOWN
- TAPI_SIM_CARD_TYPE_GSM
- TAPI_SIM_CARD_TYPE_USIM

Async Response Message:

- None

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.66 TelTapiSimPBGetStorageCount

Functions

- TapiResult_t TelTapiSimPBGetStorageCount (TelSimPbFileType_t StorageFileType, int ∗pRequestId)

  This API is used to get number of used records and total records in SIM phone book files. This API is applicable only to GSM phonebook (ADN and EXT1), FDN, SDN, LND and MSISDN.

5.66.1 Function Documentation

5.66.1.1 TapiResult_t TelTapiSimPBGetStorageCount (TelSimPbFileType_t *StorageFileType*, int ∗ *pRequestId*)

This API is used to get number of used records and total records in SIM phone book files. This API is applicable only to GSM phonebook (ADN and EXT1), FDN, SDN, LND and MSISDN.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- This API is applicable only for 2G phonebook.

Warning:

- None.

Parameters:

← ***StorageFileType*** -Different storage types to be selected in SIM.

→ ***pRequestId***
- Unique identifier for a particular request.
  - request_id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF and the event data is TelSimPbStorageInfo_t. Asynchronous return status is indicated by TelSimPbAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.67 TelTapiSimPBGetEntryInfo

Functions

- TapiResult_t TelTapiSimPBGetEntryInfo (TelSimPbFileType_t StorageFileType, int ∗pRequestId)
  This API is used to get max text length and max number length supported by SIM phone book elementary file.

5.67.1 Function Documentation

5.67.1.1 TapiResult_t TelTapiSimPBGetEntryInfo (TelSimPbFileType_t *StorageFileType*, int ∗ *pRequestId*)

This API is used to get max text length and max number length supported by SIM phone book elementary file.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- The max number length includes the storage space provided by the corresponding EXT file for a given Dialling Number file.
- This API is applicable only for 2G phonebook.

Warning:

- None.

Parameters:

← ***StorageFileType*** -Different storage types to be selected in SIM.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF and the event data is TelSimPbEntryInfo_t. Asynchronous return status is indicated by TelSimPbAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.68 TelTapiSimPBReadRecord

Functions

- TapiResult_t TelTapiSimPBReadRecord (TelSimPbFileType_t StorageFileType, unsigned short Index, int ∗pRequestId)

  This API is used to read SIM phone book entry information from given storage type.

5.68.1 Function Documentation

5.68.1.1 TapiResult_t TelTapiSimPBReadRecord (TelSimPbFileType_t *StorageFileType*, unsigned short *Index*, int ∗ *pRequestId*)

This API is used to read SIM phone book entry information from given storage type.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- The index ranges from 1 through to a maximum of 254 for a Linear fixed file and 255 for a cyclic file.
- This API is applicable only for 2G phonebook.

Warning:

- None.

Parameters:

← ***StorageFileType*** -Different storage types to be selected in SIM.

← ***Index*** -Index for accessing the SIM data.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_PB_ACCESS_READ_CNF and the event data is TelSimPbReadResp_t. Asynchronous return status is indicated by TelSimPbAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.69 TelTapiSimPBUpdateRecord

Functions

- TapiResult_t TelTapiSimPBUpdateRecord (const TelSimPbRecordData_t ∗pPbRecordData, int ∗pRequestId)

  This API is used to add or edit SIM phone book record entry information.

5.69.1 Function Documentation

5.69.1.1 TapiResult_t TelTapiSimPBUpdateRecord (const TelSimPbRecordData_t ∗ *pPbRecordData*, int ∗ *pRequestId*)

This API is used to add or edit SIM phone book record entry information.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- This API is applicable only for 2G phonebook.

Warning:

- None.

Parameters:

← ***pPbRecordData*** -Different storage types to be selected in SIM.

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_PB_UPDATE_CNF and the event data is TelSimPbUpdateResp_t. Asynchronous return status is indicated by TelSimPbAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.70 TelTapiSimPBDeleteRecord

Functions

- TapiResult_t TelTapiSimPBDeleteRecord (TelSimPbFileType_t StorageFileType, unsigned short RecordIndex, int ∗pRequestId)

  This API is used to delete a SIM phone book record.

5.70.1 Function Documentation

5.70.1.1 TapiResult_t TelTapiSimPBDeleteRecord (TelSimPbFileType_t *StorageFileType*, unsigned short *RecordIndex*, int ∗ *pRequestId*)

This API is used to delete a SIM phone book record.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- The index ranges from 1 through to a maximum of 254 for a Linear fixed file and 255 for a cyclic file. This API is applicable only for 2G phonebook.

Warning:

- None.

Parameters:

← ***StorageFileType*** -Different storage types to be selected in SIM.

← ***RecordIndex*** -Index of the record to be deleted.

→ ***pRequestId***
- Unique identifier for a particular request.
  - request_id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SIM_PB_DELETE_CNF and the event data is TelSimPbUpdateResp_t. Asynchronous return status is indicated by TelSimPbAccessResult_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Otherwise it provides an error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.71 SupplementaryServices

Files

- file ITapiSs.h

  This file serves as a "C" header file defines functions for Tapi Supplementary Services. It contains a sample set of function prototypes that would be required by applications.

Modules

- TelTapiSsSetCallBarring
- TelTapiSsGetCallBarringStatus
- TelTapiSsChangeBarPassword
- TelTapiSsSetCallForward
- TelTapiSsGetCallForwardStatus
- TelTapiSsSetCallWaiting
- TelTapiSsGetCallWaitingStatus
- TelTapiSsVerifyBarPwdResponse
- TelTapiSsSetRestrictCallId
- TelTapiSsGetRestrictCallId
- TelTapiSsQueryCapabilityInfo
- TelTapiUssdSendRequest
- TelTapiUssdSendUserResponse

5.72 TelTapiSsSetCallBarring

Functions

- TapiResult_t TelTapiSsSetCallBarring (const TelSsCallBarringInfo_t ∗pBarringInfo, int ∗pRequestId)

  This function sends a request to activates/deactivates call barring. This service makes it possible for a mobile subscriber to activate barring of certain categories of outgoing or incoming calls according to a barring program which is selected from a set of one or more barring programs chosen at provision time and is valid for all outgoing or incoming calls, or just those associated with a specific basic service group. Also this API allows deactivating call barring based on categories set.

5.72.1 Function Documentation

5.72.1.1 TapiResult_t TelTapiSsSetCallBarring (const TelSsCallBarringInfo_t ∗ *pBarringInfo*, int ∗ *pRequestId*)

This function sends a request to activates/deactivates call barring. This service makes it possible for a mobile subscriber to activate barring of certain categories of outgoing or incoming calls according to a barring program which is selected from a set of one or more barring programs chosen at provision time and is valid for all outgoing or incoming calls, or just those associated with a specific basic service group. Also this API allows deactivating call barring based on categories set.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pBarringInfo*** • Contains the information about call barring

→ ***pRequestId*** • Unique identifier for a particular request.

  - request_id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

An asynchronous notification event is returned as a confirmation The event associated is TAPI_EVENT_SS_BARRING_CNF and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.73 TelTapiSsGetCallBarringStatus

Functions

- TapiResult_t TelTapiSsGetCallBarringStatus (TelSsCallBarType_t BarType, TelSsCallType_t CallType, int ∗pRequestId)

  This function gets call bar status. The mobile subscriber may determine by subscription of a set of one or more unique barring programs what kind of outgoing or incoming calls should be barred.

5.73.1 Function Documentation

5.73.1.1 TapiResult_t TelTapiSsGetCallBarringStatus (TelSsCallBarType_t *BarType*, TelSsCallType_t *CallType*, int ∗ *pRequestId*)

This function gets call bar status. The mobile subscriber may determine by subscription of a set of one or more unique barring programs what kind of outgoing or incoming calls should be barred.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***BarType*** • Specifies barring type i.e. Outgoing international, outgoing all, Incoming all etc.

← ***CallType*** • Specifies the type of call i.e. Voice call, Video Call etc.

→ ***pRequestId*** • Unique identifier for a particular request.

- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.74 TelTapiSsChangeBarPassword

Functions

- TapiResult_t TelTapiSsChangeBarPassword (const char ∗pOldPassword, const char ∗pNewPassword, int ∗pRequestId)

  This function allows changing of the barring password in Network.

5.74.1 Function Documentation

5.74.1.1 TapiResult_t TelTapiSsChangeBarPassword (const char ∗ *pOldPassword*, const char ∗ *pNewPassword*, int ∗ *pRequestId*)

This function allows changing of the barring password in Network.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Based on 3GPP TS 22.030 section 6.5.4 Registration of New Password. Only one Change Barring Password request is allowed at a time and TS might return TAPI_API_REQUEST_MAX_IN_PROGRESS if another request is received.

Warning:

- None.

Parameters:

← ***pOldPassword***	• Old password set for Barring in Network. This is Null terminated string
← ***pNewPassword***	• New password for Barring to be set in Network. This is Null terminated string.
→ ***pRequestId***	• Unique identifier for a particular request. • request_id value can be any value from 0 to 255 if the API is returned successfully • -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF and asynchronous return status is indicated by TelSsCause_t. There is no event data with this event.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.75 TelTapiSsSetCallForward

Functions

- TapiResult_t TelTapiSsSetCallForward (const TelSsForwardInfo_t ∗pForwardInfo, int ∗pRequestId)

  This function allows to set (register/erase/activate/deactivate) call forwarding option at the Network. This enables to forward the calls to another party. This service permits a called mobile subscriber to have the network send all incoming calls, or just those associated with a specific Basic service group, addressed to the called mobile subscriber's directory number and which meet various call FORWARD condition to another directory number. The ability of the served mobile subscriber to originate calls is unaffected. If the service is activated, a call is forwarded only if under this call forward condition. Application has to call this API in order to explicitly register/erase/activate/deactivate the call forwarding. Request will be passed irrespective of whether already another SS call forward is ongoing.

5.75.1 Function Documentation

5.75.1.1 TapiResult_t TelTapiSsSetCallForward (const TelSsForwardInfo_t ∗ *pForwardInfo*, int ∗ *pRequestId*)

This function allows to set (register/erase/activate/deactivate) call forwarding option at the Network. This enables to forward the calls to another party. This service permits a called mobile subscriber to have the network send all incoming calls, or just those associated with a specific Basic service group, addressed to the called mobile subscriber's directory number and which meet various call FORWARD condition to another directory number. The ability of the served mobile subscriber to originate calls is unaffected. If the service is activated, a call is forwarded only if under this call forward condition. Application has to call this API in order to explicitly register/erase/activate/deactivate the call forwarding. Request will be passed irrespective of whether already another SS call forward is ongoing.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pForwardInfo*** • Contains the Call forward information like forward mode, forward type, etc.

→ ***pRequestId*** • Unique identifier for a particular request.

- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_FORWARD_CNF and asynchronous return status is indicated by TelSsCause_t. The event data will be TelSsInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

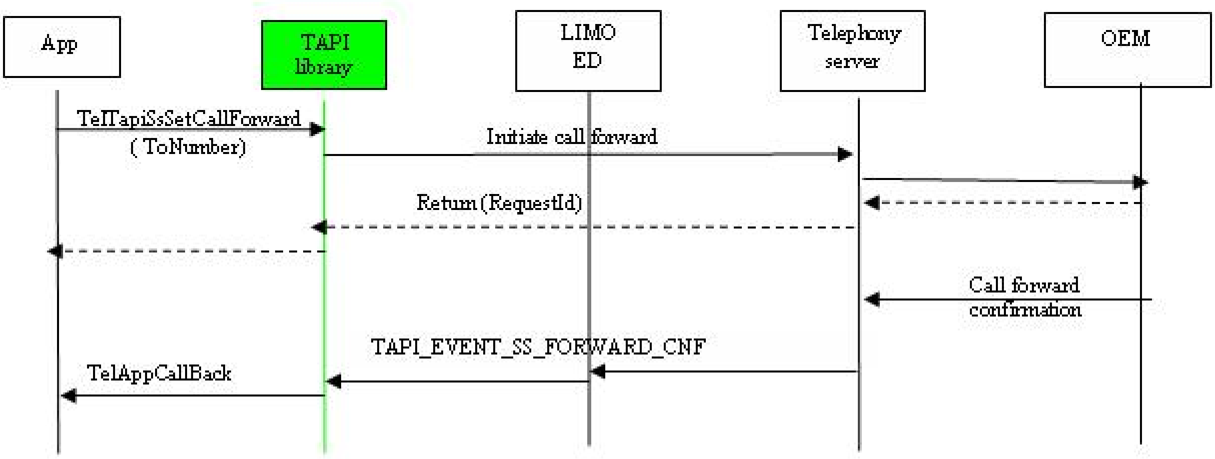
Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.



5.76 TelTapiSsGetCallForwardStatus

Functions

- TapiResult_t TelTapiSsGetCallForwardStatus (TelSsForwardType_t Type, TelSsForwardWhen_t Condition, int ∗pRequestId)

  This function provides option to get the call forwarding status of different calls from Network.

5.76.1 Function Documentation

5.76.1.1 TapiResult_t TelTapiSsGetCallForwardStatus (TelSsForwardType_t *Type*, TelSsForwardWhen_t *Condition*, int ∗ *pRequestId*)

This function provides option to get the call forwarding status of different calls from Network.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***Type***
- Specifying the Forward call type i.e. voice call, Video Call, Data Calls etc.

← ***Condition***
- Used to specify the forward condition i.e. unconditional, When no reply, when busy, when not reachable etc.

→ ***pRequestId***
- Unique identifier for a particular request.
  - request_id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.77 TelTapiSsSetCallWaiting

Functions

- TapiResult_t TelTapiSsSetCallWaiting (BOOL bCwStatus, int ∗pRequestId)

  This function activates/deactivates the call waiting service. The call waiting service permits a mobile to be notified of an incoming call (as per basic call procedures) whilst the traffic channel (BM or LM) is not available for the incoming call and the mobile subscriber is engaged in an active or held call. Subsequently, the subscriber can accept, reject, or ignore the incoming call.

5.77.1 Function Documentation

5.77.1.1 TapiResult_t TelTapiSsSetCallWaiting (BOOL *bCwStatus*, int ∗ *pRequestId*)

This function activates/deactivates the call waiting service. The call waiting service permits a mobile to be notified of an incoming call (as per basic call procedures) whilst the traffic channel (BM or LM) is not available for the incoming call and the mobile subscriber is engaged in an active or held call. Subsequently, the subscriber can accept, reject, or ignore the incoming call.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***bCwStatus***
- Specifies the status of call-waiting service i.e. On (TRUE) or off (FALSE)

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_WAITING_CNF and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.78 TelTapiSsGetCallWaitingStatus

Functions

- TapiResult_t TelTapiSsGetCallWaitingStatus (int ∗pRequestId)

  This function queries the status of call waiting service. The call waiting service permits a mobile to be notified of an incoming call (as per basic call procedures) whilst the traffic channel (BM or LM) is not available for the incoming call and the mobile subscriber is engaged in an active or held call. Subsequently, the subscriber can accept, reject, or ignore the incoming call.

5.78.1 Function Documentation

5.78.1.1 TapiResult_t TelTapiSsGetCallWaitingStatus (int ∗ *pRequestId*)

This function queries the status of call waiting service. The call waiting service permits a mobile to be notified of an incoming call (as per basic call procedures) whilst the traffic channel (BM or LM) is not available for the incoming call and the mobile subscriber is engaged in an active or held call. Subsequently, the subscriber can accept, reject, or ignore the incoming call.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
  - request_id value can be any value from 0 to 255 if the API is returned successfully
  - -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.79 TelTapiSsVerifyBarPwdResponse

Functions

- TapiResult_t TelTapiSsVerifyBarPwdResponse (int ∗pRequestId, const char ∗pBarPwdInfo)

  This API allows Application to provide barring password requested by network for Call Barring Request or Change barring password request.

5.79.1 Function Documentation

5.79.1.1 TapiResult_t TelTapiSsVerifyBarPwdResponse (int ∗ *pRequestId*, const char ∗ *pBarPwdInfo*)

This API allows Application to provide barring password requested by network for Call Barring Request or Change barring password request.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pBarPwdInfo***
- Barring password is 4 digit interger. (Character array of password digits and not a Null terminated string).

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

This Asynchronous event TAPI_EVENT_SS_VERIFY_PWD_CNF is returned in response and asynchronous return status is indicated by TelSsCause_t. There is no event data associated with this event.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.80 TelTapiSsSetRestrictCallId

Functions

- TapiResult_t TelTapiSsSetRestrictCallId (TelSsRestrictIDStatus_t Status)

  This function sets the status of restrict call ID. After set, the phone will show/hide the caller number to the called party for all subsequent calls, regardless of type of call. This API assumes user service subscription allows changing this setting, sometimes referred to as 'temporary mode'. If the presentation mode is not temporary, the caller of this API will receive an error result.

5.80.1 Function Documentation

5.80.1.1 TapiResult_t TelTapiSsSetRestrictCallId (TelSsRestrictIDStatus_t *Status*)

This function sets the status of restrict call ID. After set, the phone will show/hide the caller number to the called party for all subsequent calls, regardless of type of call. This API assumes user service subscription allows changing this setting, sometimes referred to as 'temporary mode'. If the presentation mode is not temporary, the caller of this API will receive an error result.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- None

Warning:

- None.

Parameters:

← ***Status*** • Specifies whether to restrict call id or not i.e On or Off.

Async Response Message:

- None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.81 TelTapiSsGetRestrictCallId

Functions

- TapiResult_t TelTapiSsGetRestrictCallId (TelSsRestrictIDStatus_t ∗pStatus)
  This function gets the status of call ID restriction for all subsequent calls.

5.81.1 Function Documentation

5.81.1.1 TapiResult_t TelTapiSsGetRestrictCallId (TelSsRestrictIDStatus_t ∗ *pStatus*)

This function gets the status of call ID restriction for all subsequent calls.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

→ ***pStatus***	• Specifies whether to restrict call id or not i.e On or Off.

Async Response Message:

None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.82 TelTapiSsQueryCapabilityInfo

Functions

- TapiResult_t TelTapiSsQueryCapabilityInfo (int ∗pRequestId)
  This API allows Application to query what all SS services are supported at OEM.

5.82.1 Function Documentation

5.82.1.1 TapiResult_t TelTapiSsQueryCapabilityInfo (int ∗ *pRequestId*)

This API allows Application to query what all SS services are supported at OEM.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- Note:

Warning:

- None.

Parameters:

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

This Asynchronous event TAPI_EVENT_SS_QUERYCAPABILITY_INFO_CNF is returned in response and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsCapabilityInfo_t.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.83 TelTapiUssdSendRequest

Functions

- TapiResult_t TelTapiUssdSendRequest (const TelSsUssdMsgInfo_t ∗pAppMsg, int ∗pRequestId)

  This function is used to send USSD to the Network. The response to the request is a displayable string which will be returned to the Application via LiMo EventDelivery.

5.83.1 Function Documentation

5.83.1.1 TapiResult_t TelTapiUssdSendRequest (const TelSsUssdMsgInfo_t ∗ *pAppMsg*, int ∗ *pRequestId*)

This function is used to send USSD to the Network. The response to the request is a displayable string which will be returned to the Application via LiMo EventDelivery.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pAppMsg***
- Data coding scheme used (GSM 7bit or 8bit default alphabet etc, defined as 3GPP TS 23.038)
- USSD String data length
- ANSI string passed by the client

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_USSD_CNF and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsUssdMsgInfo_t.

Pre Conditions:

- No SS and USSD Transaction should be ongoing. If there an ongoing transaction, new USSD request will be returned to the Application with an error (USSD_BUSY)

Post Conditions:

- None.

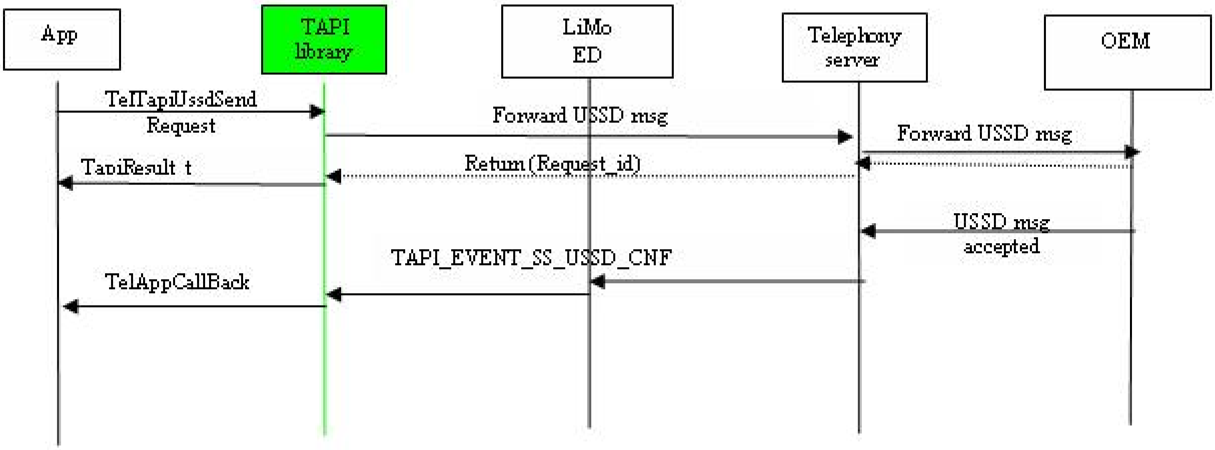
Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.



5.84 TelTapiUssdSendUserResponse

Functions

- TapiResult_t TelTapiUssdSendUserResponse (const TelSsUssdMsgInfo_t ∗pAppMsg, int ∗pRequestId)

  This function is called to send a USSD response provided by the User to the Network for the USSD indication received. This contains USSD response type, USSD Indication type and USSD Response data.

5.84.1 Function Documentation

5.84.1.1 TapiResult_t TelTapiUssdSendUserResponse (const TelSsUssdMsgInfo_t ∗ *pAppMsg*, int ∗ *pRequestId*)

This function is called to send a USSD response provided by the User to the Network for the USSD indication received. This contains USSD response type, USSD Indication type and USSD Response data.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

- None.

Parameters:

← ***pAppMsg***
- Data coding scheme used (GSM 7bit or 8bit default alphabet etc, defined as 3GPP TS 23.038)
- USSD String data length
- ANSI string passed by the client

→ ***pRequestId***
- Unique identifier for a particular request.
- request_id value can be any value from 0 to 255 if the API is returned successfully.
- -1 (INVALID_REQUEST_ID) will be sent in case of failure.

Async Response Message:

The event associated is TAPI_EVENT_SS_USSD_IND_RSP and asynchronous return status is indicated by TelSsCause_t. The event data is TelSsUssdMsgInfo_t.

Pre Conditions:

- No SS and USSD Transaction should be ongoing. If there an ongoing transaction, new USSD request will be returned to the Application with an error (USSD_BUSY)

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Refer TapiResult_t for failure and error code

Prospective Clients:

External Apps.

5.85 Common

Files

- file TapiCommon.h

  This file serves as a "C" header file defines functions for Tapi Common Interfaces like Callback Registration / De-registration.
  It contains a sample set of function prototypes that would be required by applications.

Modules

- TelTapiRegisterClass
- TelTapiRegisterEvent
- TelTapiDeRegister
- TelTapiTunnelReq
- TelTapiGetVersionInfo

Data Structures

- struct TelTapiEvent_t

Typedefs

- typedef bool(∗) TelAppCallback (const TelTapiEvent_t ∗pTapiEvent)

Enumerations

- enum TelTapiEventClass_t {

  TAPI_EVENT_NONE = -1, TAPI_EVENT_CLASS_CALL_VOICE = 0x00, TAPI_EVENT_CLASS_CALL_DATA, TAPI_EVENT_CLASS_NETTEXT,

  TAPI_EVENT_CLASS_NETWORK, TAPI_EVENT_CLASS_SS, TAPI_EVENT_CLASS_SIM, TAPI_EVENT_CLASS_ERR,

  TAPI_EVENT_CLASS_UTIL, TAPI_EVENT_CLASS_ALL }
- enum TelTapiEventType_t {

  TAPI_EVENT_CALL_SETUP_CNF = 1, TAPI_EVENT_CALL_ALERT_IND, TAPI_EVENT_CALL_CONNECTING_IND, TAPI_EVENT_CALL_CONNECTED_IND,

  TAPI_EVENT_CALL_INCOM_IND, TAPI_EVENT_CALL_ANSWER_CNF, TAPI_EVENT_CALL_RETRIEVE_CNF, TAPI_EVENT_CALL_RETRIEVE_IND,

  TAPI_EVENT_CALL_HOLD_IND, TAPI_EVENT_CALL_HOLD_CNF, TAPI_EVENT_CALL_TRANSFER_CNF, TAPI_EVENT_CALL_TRANSFER_IND,

  TAPI_EVENT_CALL_SETUPCONFERENCE_CNF, TAPI_EVENT_CALL_SETUPCONFERENCE_IND, TAPI_EVENT_CALL_SPLITCONFERENCE_CNF, TAPI_EVENT_CALL_SEND_DTMF_CNF,

  TAPI_EVENT_CALL_WAITING_IND, TAPI_EVENT_CALL_FORWARD_IND, TAPI_EVENT_CALL_USERINFO_IND, TAPI_EVENT_CALL_RELEASE_CNF, TAPI_EVENT_CALL_RELEASE_ALL_CNF, TAPI_EVENT_CALL_END_IND, TAPI_EVENT_NETTEXT_READ_SMS_CNF, TAPI_EVENT_NETTEXT_GET_COUNT_CNF,

TAPI_EVENT_NETTEXT_SENTSTATUS_CNF, TAPI_EVENT_NETTEXT_GET_PARAM_CNF, TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF, TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF,

TAPI_EVENT_NETTEXT_SET_REQUEST_CNF, TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF, TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF, TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF,

TAPI_EVENT_NETTEXT_INCOM_IND, TAPI_EVENT_NETTEXT_CB_INCOM_IND, TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND, TAPI_EVENT_NETWORK_SELECT_CNF,

TAPI_EVENT_NETWORK_SEARCH_CNF, TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF, TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF, TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF,

TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF, TAPI_EVENT_NETWORK_SETNWORDER_CNF, TAPI_EVENT_NETWORK_SETNWMODE_CNF, TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF,

TAPI_EVENT_NETWORK_SETNWBAND_CNF, TAPI_EVENT_NETWORK_SETSERVICEDOMAIN_CNF, TAPI_EVENT_NETWORK_CONFIG_CNF, TAPI_EVENT_NETWORK_CHANGE_IND,

TAPI_EVENT_NETWORK_SEARCH_IND, TAPI_EVENT_NETWORK_RADIO_INFO_IND, TAPI_EVENT_SS_BARRING_CNF, TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF,

TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF, TAPI_EVENT_SS_FORWARD_CNF, TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF, TAPI_EVENT_SS_WAITING_CNF,

TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF, TAPI_EVENT_SS_USSD_CNF, TAPI_EVENT_SS_USSDCNF_RSP, TAPI_EVENT_SS_USSDCNF_NOTIFY,

TAPI_EVENT_SS_USSD_IND_RSP, TAPI_EVENT_SS_VERIFY_PWD_CNF, TAPI_EVENT_SS_QUERYCAPABILITY_INFO_CNF, TAPI_EVENT_SS_USSD_IND,

TAPI_EVENT_SS_USSD_IND_NOTIFY, TAPI_EVENT_SS_GET_PASSWORD_IND, TAPI_EVENT_SIM_FILE_INFO_CNF, TAPI_EVENT_SIM_READ_FILE_CNF,

TAPI_EVENT_SIM_UPDATE_FILE_CNF, TAPI_EVENT_SIM_VERIFY_PIN_CNF, TAPI_EVENT_SIM_CHANGE_PIN_CNF, TAPI_EVENT_SIM_DISABLE_PIN_CNF,

TAPI_EVENT_SIM_ENABLE_PIN_CNF, TAPI_EVENT_SIM_UNBLOCK_PIN_CNF, TAPI_EVENT_SIM_DISABLE_FDN_CNF, TAPI_EVENT_SIM_ENABLE_FDN_CNF,

TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF, TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF, TAPI_EVENT_SIM_PB_ACCESS_READ_CNF, TAPI_EVENT_SIM_PB_UPDATE_CNF,

TAPI_EVENT_SIM_PB_DELETE_CNF, TAPI_EVENT_SIM_STATUS_IND, TAPI_EVENT_SIM_FDN_STATUS_IND, TAPI_EVENT_ERR_RESERVED1_IND,

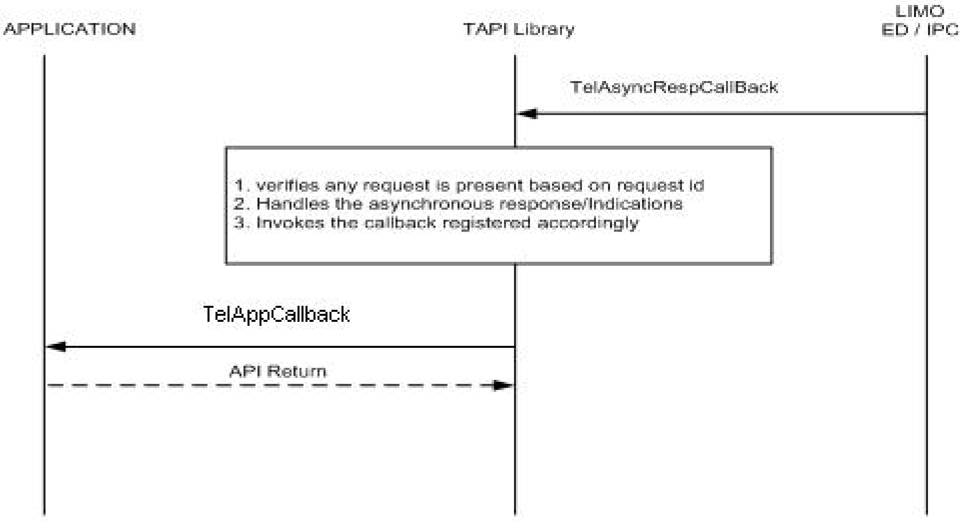
TAPI_EVENT_ERR_RESERVED2_IND, TAPI_EVENT_ERR_RESERVED3_IND, TAPI_EVENT_ERR_RESERVED4_IND, TAPI_EVENT_ERR_RESERVED5_IND,

TAPI_EVENT_TUNNEL_INFO_CNF, TAPI_EVENT_TUNNEL_INFO_IND }

5.85.1 Typedef Documentation

5.85.1.1 typedef bool(∗) TelAppCallback(const TelTapiEvent_t ∗pTapiEvent)

This describes the application callback function which will get invoked to notify the client application about the Telephony events based on TAPI asynchronous responses / indications.


APPLICATION
TAPI Library
LIMO
ED / IPC
TelAsyncRespCallBack
1. verifies any request is present based on request id
2. Handles the asynchronous response/Indications
3. Invokes the callback registered accordingly
TelAppCallback
API Return

5.85.2 Enumeration Type Documentation

5.85.2.1 enum TelTapiEventClass_t

This enumeration defines the Telephony Event Class. Class is nothing but a group of events related to a particular subsystem. Example of one Event Class is TAPI_CALL_VOICE. This becomes the first parameter of telephony notifications.

Enumerator:

TAPI_EVENT_CLASS_CALL_VOICE	Voice Call Event Class
TAPI_EVENT_CLASS_CALL_DATA	Data Call Event Class
TAPI_EVENT_CLASS_NETTEXT	Nettext (SMS/CB) Event Class
TAPI_EVENT_CLASS_NETWORK	Network Event Class
TAPI_EVENT_CLASS_SS	SS Event Class
TAPI_EVENT_CLASS_SIM	SIM Event Class
TAPI_EVENT_CLASS_ERR	Fatal error Event Class
TAPI_EVENT_CLASS_UTIL	Utility Events Class
TAPI_EVENT_CLASS_ALL	All Events Class

5.85.2.2 enum TelTapiEventType_t

This enumeration defines the event types. Each event type will be mapped to one telephony Notification Id. For Example: TAPI_EVENT_CALL_INCOM_IND is mapped to Telephony.Call.Voice.Incoming.

Enumerator:

TAPI_EVENT_CALL_SETUP_CNF	This event indicates the Call Setup request triggered
TAPI_EVENT_CALL_ALERT_IND	This event indicates the Call is in Alerting stage
TAPI_EVENT_CALL_CONNECTING_IND	This event indicates the Call is in Connecting stage
TAPI_EVENT_CALL_CONNECTED_IND	This event indicates the Call is Connected

TAPI_EVENT_CALL_INCOM_IND This event indicates Incoming Call indication to Application

TAPI_EVENT_CALL_ANSWER_CNF Incoming call was answered

TAPI_EVENT_CALL_RETRIEVE_CNF This event indicates Call is retrieved from held state

TAPI_EVENT_CALL_RETRIEVE_IND This event indicates that other party has retrieved the call from hold

TAPI_EVENT_CALL_HOLD_IND This event indicates that other party is has put the call on hold

TAPI_EVENT_CALL_HOLD_CNF This event indicates is call is put on hold

TAPI_EVENT_CALL_TRANSFER_CNF This event indicates is call is explicitly transferred

TAPI_EVENT_CALL_TRANSFER_IND Call Transfer Indication

TAPI_EVENT_CALL_SETUPCONFERENCE_CNF This event indicates that Call is joined in a conference

TAPI_EVENT_CALL_SETUPCONFERENCE_IND Conference setup indication

TAPI_EVENT_CALL_SPLITCONFERENCE_CNF This event indicates Call is split from conference

TAPI_EVENT_CALL_SEND_DTMF_CNF This event indicates the confirmation for sending DTMF Digits to the network

TAPI_EVENT_CALL_WAITING_IND Call waiting indication

TAPI_EVENT_CALL_FORWARD_IND This event indication for Call is forwarded

TAPI_EVENT_CALL_USERINFO_IND This event is indicated when User Information is exchanged during the call

TAPI_EVENT_CALL_RELEASE_CNF This event is indicated when all the calls released requested by the application

TAPI_EVENT_CALL_RELEASE_ALL_CNF This event is indicated when all the calls released requested by the application

TAPI_EVENT_CALL_END_IND This event indicates the Call is Ended. This event can be solicited/unsolicited.

TAPI_EVENT_NETTEXT_READ_SMS_CNF This event is returned in response to SMS Read message

TAPI_EVENT_NETTEXT_GET_COUNT_CNF This event is returned in response to Get Number of messages stored in SIM

TAPI_EVENT_NETTEXT_SENTSTATUS_CNF This event is returned in response to SMS message sent

TAPI_EVENT_NETTEXT_GET_PARAM_CNF This event is returned in response to NetText Get parameters

TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF This confirmation event is a returned indicating the delete message e status

TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF This confirmation event is a returned indicating the Save SMS into SIM status

TAPI_EVENT_NETTEXT_SET_REQUEST_CNF This event is returned as a confirmation for SET nettext APIs

TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF This event is returned with Get CB configuration details

TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF This event is returned as a response to get preferred bearer used for SMS Transport

TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF This event is returned in response to send delivery report

TAPI_EVENT_NETTEXT_INCOM_IND This is an unsolicited Incoming SMS Event

TAPI_EVENT_NETTEXT_CB_INCOM_IND This is an unsolicited Incoming cell broadcast message

TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND This event is an unsolicited event indicating SIM memory status

TAPI_EVENT_NETWORK_SELECT_CNF This confirmation event is returned when user manually selected a network from list for Network selection

TAPI_EVENT_NETWORK_SEARCH_CNF This event is returned as confirmation for manual Network search with the available network list

TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF This event is returned when Get selection mode indication is received at TS

TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF This event is returned when Get network information is requested

TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF This event is returned on cancellation of manual network search

TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF This event is returned on cancellation of manual network selection from network list

TAPI_EVENT_NETWORK_SETNWORDER_CNF This event is returned on setting network order

TAPI_EVENT_NETWORK_SETNWMODE_CNF This event is returned on setting network mode

TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF This event is returned on setting network selection mode

TAPI_EVENT_NETWORK_SETNWBAND_CNF This event is returned on setting network band

TAPI_EVENT_NETWORK_SETSERVICEDOMAIN_CNF This event is returned on setting network service domain

TAPI_EVENT_NETWORK_CONFIG_CNF This event is returned on setting the network configuration at OEM

TAPI_EVENT_NETWORK_CHANGE_IND This event is returned when Telephony server receives any indication from OEM about change in Network registration/selection/status

TAPI_EVENT_NETWORK_SEARCH_IND This is unsolicited event indicated when TS receives any indication of Manual search indication with list

TAPI_EVENT_NETWORK_RADIO_INFO_IND This is unsolicited event indicating Radio signal indication

TAPI_EVENT_SS_BARRING_CNF This confirmation event is sent when Barring activation/deactivation is requested

TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF This event is returned as Barring password change confirmation

TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF This event is returned as Barring status query confirmation

TAPI_EVENT_SS_FORWARD_CNF This event is returned as confirmation for Forward Activation/Deactivation/Registration/Erasure

TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF This event is returned as confirmation for Forward Status query confirmation

TAPI_EVENT_SS_WAITING_CNF This event is returned as confirmation for Call waiting activation/deactivation

TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF This event is returned as Call waiting status query confirmation

TAPI_EVENT_SS_USSD_CNF This event is returned as USSD send confirmation

TAPI_EVENT_SS_USSDCNF_RSP This event is returned as USSD confirmation response which is sent as a response to received USSD

TAPI_EVENT_SS_USSDCNF_NOTIFY This event is an unsolicited USSD notification

TAPI_EVENT_SS_USSD_IND_RSP USSD Indication Response

TAPI_EVENT_SS_VERIFY_PWD_CNF This event is sent when verify pwd response is sent to Network

TAPI_EVENT_SS_QUERYCAPABILITY_INFO_CNF Event associated with the interrogation of supported supplementary services at OEM.

TAPI_EVENT_SS_USSD_IND This is an unsolicited USSD indication

TAPI_EVENT_SS_USSD_IND_NOTIFY This event is an unsolicited USSD notification

TAPI_EVENT_SS_GET_PASSWORD_IND This event is indicated to the Application when network requests for Call barring Get password for barring operation

TAPI_EVENT_SIM_FILE_INFO_CNF File info

TAPI_EVENT_SIM_READ_FILE_CNF Read file

TAPI_EVENT_SIM_UPDATE_FILE_CNF Updated file

TAPI_EVENT_SIM_VERIFY_PIN_CNF Verified Chv

TAPI_EVENT_SIM_CHANGE_PIN_CNF Changed Chv

TAPI_EVENT_SIM_DISABLE_PIN_CNF Disabled Chv

TAPI_EVENT_SIM_ENABLE_PIN_CNF Enabled Chv

TAPI_EVENT_SIM_UNBLOCK_PIN_CNF Unblocked Chv

TAPI_EVENT_SIM_DISABLE_FDN_CNF Disabled Fdn

TAPI_EVENT_SIM_ENABLE_FDN_CNF Enabled Fdn

TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF Phonebook storage count

TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF Phonebook entry information

TAPI_EVENT_SIM_PB_ACCESS_READ_CNF Phonebook access read

TAPI_EVENT_SIM_PB_UPDATE_CNF Phonebook update result

TAPI_EVENT_SIM_PB_DELETE_CNF Phonebook delete result

TAPI_EVENT_SIM_STATUS_IND Unsolicited event — whenever there is a changes in SIM status

TAPI_EVENT_SIM_FDN_STATUS_IND Unsolicited event — whenever there is a changes in FDN status

TAPI_EVENT_ERR_RESERVED1_IND This event is notified to application about some error has occurred at OEM and can be used for debugging purpose

TAPI_EVENT_ERR_RESERVED2_IND This event is notified to application about some error has occurred at OEM and can be used for debugging purpose

TAPI_EVENT_ERR_RESERVED3_IND This event is notified to application about some error has occurred at OEM and can be used for debugging purpose

TAPI_EVENT_ERR_RESERVED4_IND This event is notified to application about some error has occurred at OEM and can be used for debugging purpose

TAPI_EVENT_ERR_RESERVED5_IND This event is notified to application about some error has occurred at OEM and can be used for debugging purpose

TAPI_EVENT_TUNNEL_INFO_CNF This confirmation event is returned for Tunnel information

TAPI_EVENT_TUNNEL_INFO_IND This event is an unsolicited event informing about Tunnel information

5.86 TelTapiRegisterClass

Functions

- TapiResult_t TelTapiRegisterClass (TelTapiEventClass_t EventClass, unsigned int ∗pSubscriptionId, TelAppCallback AppCb)

  This function is exposed to the TAPI clients to register the callback which needs to be invoked for a particular class of Telephony confirmations/indications. If Application tries to call this TelTapiRegisterClass API for the same event class multiple times (with same callback or different callback), multiple callback registration will happen. SubscriptionID is returned back to the application, if the registration procedure is successful. This subscription id needs to be used for de-registering the callback.

5.86.1 Function Documentation

5.86.1.1 TapiResult_t TelTapiRegisterClass (TelTapiEventClass_t *EventClass*, unsigned int ∗ *pSubscriptionId*, TelAppCallback *AppCb*)

This function is exposed to the TAPI clients to register the callback which needs to be invoked for a particular class of Telephony confirmations/indications. If Application tries to call this TelTapiRegisterClass API for the same event class multiple times (with same callback or different callback), multiple callback registration will happen. SubscriptionID is returned back to the application, if the registration procedure is successful. This subscription id needs to be used for de-registering the callback.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- Incase Application has registered for multiple callbacks for an event class and it wants to have only one callback to be registered for that particular event class, then Application needs to explicitly deregister the callbacks based on subscription id. If Application wants to modify the already registered event class with a different callback, then it needs to deregister the already registered TAPI event class with subscription id received and Application has to register again with a new callback for that TAPI event class.

Warning:

Parameters:

← ***EventClass***	• EventClass for which the callback registration is required.
← ***AppCb***	• Callback which will be invoked when an event occurs for that particular event class.
→ ***pSubscriptionId***	• An unique identifier identifying the callback registration.

Async Response Message:

- None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.87 TelTapiRegisterEvent

Functions

- TapiResult_t TelTapiRegisterEvent (TelTapiEventType_t EventType, unsigned int ∗pSubscriptionId, TelAppCallback AppCb)

  This function is used to register the callback function for a specific TAPI Event. If Application tries to register a callback for a specific event multiple times, then multiple callbacks will be registered. When an event indication is received, then registered callbacks for the specific event will be invoked multiple times.

5.87.1 Function Documentation

5.87.1.1 TapiResult_t TelTapiRegisterEvent (TelTapiEventType_t *EventType*, unsigned int ∗ *pSubscriptionId*, TelAppCallback *AppCb*)

This function is used to register the callback function for a specific TAPI Event. If Application tries to register a callback for a specific event multiple times, then multiple callbacks will be registered. When an event indication is received, then registered callbacks for the specific event will be invoked multiple times.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- Incase Application has registered for multiple callbacks for a event and it wants to have only one callback to be registered for that particular event, then Application needs to explicitly deregister the callbacks based on subscription id. If Application wants to modify the already registered event with a different callback, then it needs to deregister the already registered TAPI event with subscription id received and Application has to register again with a new callback for that TAPI event.

Warning:

Parameters:

← ***EventType***	• TAPI Event for which the callback registration is required.
← ***AppCb***	• Callback which will be invoked when an event occurs for that particular event class.
→ ***pSubscriptionId***	• An unique identifier identifying the callback registration.

Async Response Message:

- None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.88 TelTapiDeRegister

Functions

- TapiResult_t TelTapiDeRegister (unsigned int SubscriptionId)

  This function will de-register the application callback registered for the specific event or event class based on the subscription id. This identifier is the same subscription identifier which is returned back to the application during TAPI registration procedure. If de-register API is called during any service requests, Application registered callback will be cleared based on the subscription id. No application callback shall be invoked in case any confirmations/indications are received from TS.

5.88.1 Function Documentation

5.88.1.1 TapiResult_t TelTapiDeRegister (unsigned int *SubscriptionId*)

This function will de-register the application callback registered for the specific event or event class based on the subscription id. This identifier is the same subscription identifier which is returned back to the application during TAPI registration procedure. If de-register API is called during any service requests, Application registered callback will be cleared based on the subscription id. No application callback shall be invoked in case any confirmations/indications are received from TS.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- Application has to de-register each registration using subscription id explicitly as there is no overriding of callback registration.

Warning:

Parameters:

← ***SubscriptionId***	• An unique identifier identifying the callback registration.

Async Response Message:

- None.

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.

- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.89 TelTapiTunnelReq

Functions

- TapiResult_t TelTapiTunnelReq (const TelTapiTunnelInfo_t ∗pTunnelInfo, int ∗pRequestId)
  This API allows Application to send any proprietary information to OEM.

5.89.1 Function Documentation

5.89.1.1 TapiResult_t TelTapiTunnelReq (const TelTapiTunnelInfo_t ∗ *pTunnelInfo*, int ∗ *pRequestId*)

This API allows Application to send any proprietary information to OEM.

Sync (or) Async:

This is an Asynchronous API.

Important Notes:

- None.

Warning:

Parameters:

← ***pTunnelInfo*** • This information holds the length of the tunneling data and actual tunneling data.

→ ***pRequestId*** • Callback which will be invoked when an event occurs for that particular event class.

Async Response Message:

- This Asynchronous event TAPI_EVENT_TUNNEL_INFO_CNF is returned in response and associated event data is TelTapiTunnelInfo_t with this event. Asynchronous return status will be indicated by tunnel_status which is a Boolean indicating success(1-True) or Failure(0-False)

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.90 TelTapiGetVersionInfo

Functions

- TapiResult_t TelTapiGetVersionInfo (const TelTapiVersionInfo_t *pVersionInfo)
  This function returns the version information of Telephony Server.

5.90.1 Function Documentation

5.90.1.1 TapiResult_t TelTapiGetVersionInfo (const TelTapiVersionInfo_t * *pVersionInfo*)

This function returns the version information of Telephony Server.

Sync (or) Async:

This is a Synchronous API.

Important Notes:

- None.

Warning:

Parameters:

→ ***pVersionInfo***

- Version information will be returned from Telephony Server
- This structure will hold Major version and Minor version details. For example, if Telephony version is 1.10, Major version will hold value 1 and Minor Version will hold value 10.

Async Response Message:

- NA

Pre Conditions:

- None.

Post Conditions:

- None.

Returns:

Return Type (TapiResult_t)

- TAPI_API_SUCCESS - indicating that the operation has completed successfully.
- Else it will return failure and error code (Refer TapiResult_t)

Prospective Clients:

External Apps.

5.91 Call Data Structures

Files

- file TelCall.h

  This file serves as a "C" header file defines structures for Tapi Call Services.
  It contains a sample set of constants, enums, structs that would be required by applications.

Data Structures

- struct TelCallCugInfo_t
- struct TelCallingNameInfo_t
- struct TelCallUusInfo_t
- struct TelCallSetupParams_t
- struct TelCallStatus_t
- struct TelCallUserDataInfo_t
- struct TelCallEndInfo_t
- struct TelCallRedirectionInfo_t
- struct TelCallIncomingCallInfo_t

Defines

- #define TAPI_CALLING_NAME_SIZE_MAX 80
- #define TAPI_CALL_UUS_DATA_LEN_MAX 131

Enumerations

- enum TelCallCause_t {

  TAPI_CAUSE_SUCCESS, TAPI_CAUSE_ACCESSDISCARD, TAPI_CAUSE_BARR_BYOPERATOR, TAPI_CAUSE_BEARERMODE_NOTIMPL,

  TAPI_CAUSE_BEARERMODE_RESTRICTED, TAPI_CAUSE_BEARERMODE_UNAUTH, TAPI_CAUSE_BEARERMODE_UNAVAIL, TAPI_CAUSE_BUSY,

  TAPI_CAUSE_CALLMETER_EXPIRED, TAPI_CAUSE_CALLNO_ERROR, TAPI_CAUSE_CKTUNAVAIL, TAPI_CAUSE_CONGESTION,

  TAPI_CAUSE_NO_CIRCUIT_AVAIL, TAPI_CAUSE_DESTIN_INCOMPAT, TAPI_CAUSE_DTMF_NOSPEECH, TAPI_CAUSE_DTMF_REJECTED,

  TAPI_CAUSE_FACILITY_NOTIMPL, TAPI_CAUSE_FACILITY_NOTSUBSCRIBED, TAPI_CAUSE_INCOMINGCUGCALLS_BARRED, TAPI_CAUSE_INVALNUM,

  TAPI_CAUSE_MPTY_ERROR, TAPI_CAUSE_NOANSWER, TAPI_CAUSE_NONCUGMEMBER, TAPI_CAUSE_NUMBERCHANGED,

  TAPI_CAUSE_NUMBER_ERROR, TAPI_CAUSE_NWFAILURE, TAPI_CAUSE_OPERATIONUNAVAIL, TAPI_CAUSE_OUTOFORDER,

  TAPI_CAUSE_PROTOCOL_ERROR, TAPI_CAUSE_QOSUNAVAIL, TAPI_CAUSE_REJECT, TAPI_CAUSE_REJ_FAIL,

  TAPI_CAUSE_REJ_SRVC_NOT_AVL, TAPI_CAUSE_REMOTE_CKTUNAVAIL, TAPI_CAUSE_RESOURCEUNAVAIL, TAPI_CAUSE_SERVICEID_ERROR, TAPI_CAUSE_SERVICE_NOTIMPL, TAPI_CAUSE_SERVICE_UNAVAIL, TAPI_CAUSE_MODIFY_SAME_BEARER, TAPI_CAUSE_MODIFY_NOT_PERMITTED,

TAPI_CAUSE_HOLD_FAIL, TAPI_CAUSE_TEMPFAILURE, TAPI_CAUSE_WRONGNUM, TAPI_CAUSE_NORMAL_UNSPECIFIED,

TAPI_CAUSE_SSERRINVALIDTIVALUE, TAPI_CAUSE_SSERRINVALIDTI, TAPI_CAUSE_SSINCOMPATIBLESTATE, TAPI_CAUSE_SSERRSYNTAXERROR,

TAPI_CAUSE_SSPROTOCOLERROR, TAPI_CAUSE_SSNEGATIVEPASSWORDCHECK, TAPI_CAUSE_SSSENTASPHASE1, TAPI_CAUSE_SSERROR,

TAPI_CAUSE_SS_USSD_BUSY, TAPI_CAUSE_IDLE, TAPI_CAUSE_NETWORK_SPECIFIC, TAPI_CAUSE_FADE,

TAPI_CAUSE_UNKNOWN, TAPI_CAUSE_INCOM_CALL, TAPI_CAUSE_ALERT_STOP, TAPI_CAUSE_INTERCEPT,

TAPI_CAUSE_REORDER, TAPI_CAUSE_CLIENT_END, TAPI_CAUSE_ACCESS_CLASS_BLOCKED, TAPI_CAUSE_MEMCAPEXCEEDED,

TAPI_CAUSE_TYPENOTSUPPORTED, TAPI_CAUSE_REPLACENOTSUPPORTED, TAPI_CAUSE_PROTOCOLID_ERROR, TAPI_CAUSE_CLASSNOTSUPPORTED,

TAPI_CAUSE_DATACODING_ERROR, TAPI_CAUSE_INVAL_MSG, TAPI_CAUSE_CALL_BARRED, TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_SS,

TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_CALL, TAPI_CAUSE_SIM_ERROR, TAPI_CAUSE_SIM_CALL_CONTROL_MODIFIED, TAPI_CAUSE_RANDOM_ACCESS_FAILURE,

TAPI_CAUSE_LOWER_LAYER_FAILURE, TAPI_CAUSE_ACCESS_REJECTED, TAPI_CAUSE_REEST_NOT_ALLOWED, TAPI_CAUSE_EMERGENCY_CALL_NOT_ALLOWED,

TAPI_CAUSE_ACCESS_NOT_ALLOWED, TAPI_CAUSE_RADIO_LINK_FAILURE, TAPI_CAUSE_INVALID_IDENTITY, TAPI_CAUSE_UNKNOWN_IDENTITY,

TAPI_CAUSE_UNACCEPTABLE_IDENTITY, TAPI_CAUSE_NO_SERVICE_HERE, TAPI_CAUSE_SERVICE_NOT_ALLOWED, TAPI_CAUSE_SERVICE_NOT_AVAIL,

TAPI_CAUSE_SERVICE_NOT_WORKING, TAPI_CAUSE_CANNOT_IDENTIFY_CALL, TAPI_CAUSE_DEACTIVATION, TAPI_CAUSE_FATAL_ERROR }

- enum TelCallAnswerType_t { TAPI_CALL_ANSWER_ACCEPT = 0, TAPI_CALL_ANSWER_REJECT = 1 }
- enum TelCallType_t { TAPI_CALL_TYPE_VOICE, TAPI_CALL_TYPE_DATA, TAPI_CALL_TYPE_E911 }
- enum TelCallAlertingPattern_t {

  TAPI_CALL_ALERTING_PATTERN_1 = 0x00, TAPI_CALL_ALERTING_PATTERN_2 = 0x01, TAPI_CALL_ALERTING_PATTERN_3 = 0x02, TAPI_CALL_ALERTING_PATTERN_4 = 0x03,

  TAPI_CALL_ALERTING_PATTERN_5 = 0x04, TAPI_CALL_ALERTING_PATTERN_6 = 0x05, TAPI_CALL_ALERTING_PATTERN_7 = 0x06, TAPI_CALL_ALERTING_PATTERN_8 = 0x07,

  TAPI_CALL_NO_ALERTING_PATTERN = 0x08 }
- enum TelCallActiveState_t { TAPI_CALL_CONNECT_IDLE, TAPI_CALL_CONNECT_ACTIVE, TAPI_CALL_CONNECT_HELD }
- enum TelCallIdentityMode_t { TAPI_CALL_IDENTITY_DEFAULT, TAPI_CALL_IDENTITY_SHOW, TAPI_CALL_IDENTITY_HIDE }
- enum TelCallCugOptions_t { TAPI_CALL_CUG_NO_INFO = 0x00, TAPI_CALL_CUG_SUPRESS_OA, TAPI_CALL_CUG_SUPRESS_PRF_CUG, TAPI_CALL_CUG_SUPRESS_OA_AND_CUG }
- enum TelCallUusType_t { TAPI_CALL_UUS_NONE, TAPI_CALL_UUS_1 }

- enum TelCallUusProtocolType_t {

  TAPI_CALL_UUS_PROTO_SPECIFIC = 0, TAPI_CALL_UUS_PROTO_OSI_HIGHER_LAYER = 1, TAPI_CALL_UUS_PROTO_IA5_CHARS = 4, TAPI_CALL_UUS_PROTO_V120 = 7,

  TAPI_CALL_UUS_PROTO_Q931 = 8 }
- enum TelCallNameMode_t { TAPI_CALL_NAME_AVAIL = 0, TAPI_CALL_NAME_RESTRICTED = 1, TAPI_CALL_NAME_UNAVAIL = 2, TAPI_CALL_NAME_AVAIL_RESTRICTED = 3 }

5.91.1 Define Documentation

5.91.1.1 #define TAPI_CALL_UUS_DATA_LEN_MAX 131

The Maximum length of the user to user signaling data string.

5.91.1.2 #define TAPI_CALLING_NAME_SIZE_MAX 80

The maximum length of the string for calling party name.

5.91.2 Enumeration Type Documentation

5.91.2.1 enum TelCallActiveState_t

State of the current call.

Enumerator:

TAPI_CALL_CONNECT_IDLE	Call is in Idle state.
TAPI_CALL_CONNECT_ACTIVE	Call is in Active state.
TAPI_CALL_CONNECT_HELD	Call is in Held state.

5.91.2.2 enum TelCallAlertingPattern_t

This enumeration defines the different alerting patterns available.

Enumerator:

TAPI_CALL_ALERTING_PATTERN_1	Pattern 1
TAPI_CALL_ALERTING_PATTERN_2	Pattern 2
TAPI_CALL_ALERTING_PATTERN_3	Pattern 3
TAPI_CALL_ALERTING_PATTERN_4	Pattern 4
TAPI_CALL_ALERTING_PATTERN_5	Pattern 5
TAPI_CALL_ALERTING_PATTERN_6	Pattern 6
TAPI_CALL_ALERTING_PATTERN_7	Pattern 7
TAPI_CALL_ALERTING_PATTERN_8	Pattern 8
TAPI_CALL_NO_ALERTING_PATTERN	No Alerting Pattern

5.91.2.3 enum TelCallAnswerType_t

Answer type used as in parameter in the answer call API.

Enumerator:

TAPI_CALL_ANSWER_ACCEPT	Used to answer an incoming call when there are no current active calls.
TAPI_CALL_ANSWER_REJECT	Used to reject the incoming call

5.91.2.4 enum TelCallCause_t

This structure defines the values for Call Cause.

Enumerator:

TAPI_CAUSE_SUCCESS	Success
TAPI_CAUSE_ACCESSDISCARD	Access discarded
TAPI_CAUSE_BARR_BYOPERATOR	Barred by operator
TAPI_CAUSE_BEARERMODE_NOTIMPL	Bearer mode not implemented
TAPI_CAUSE_BEARERMODE_RESTRICTED	Bearer mode restricted
TAPI_CAUSE_BEARERMODE_UNAUTH	Bearer mode un authorized
TAPI_CAUSE_BEARERMODE_UNAVAIL	Bearer mode not available
TAPI_CAUSE_BUSY	(Network/Server) busy
TAPI_CAUSE_CALLMETER_EXPIRED	Call meter expired
TAPI_CAUSE_CALLNO_ERROR	Call number error
TAPI_CAUSE_CKTUNAVAIL	Circuit channel unavailable
TAPI_CAUSE_CONGESTION	Congestion happened
TAPI_CAUSE_NO_CIRCUIT_AVAIL	Circuit not available
TAPI_CAUSE_DESTIN_INCOMPAT	Destination incompatibility
TAPI_CAUSE_DTMF_NOSPEECH	No speech in DTMF
TAPI_CAUSE_DTMF_REJECTED	DTMF rejected
TAPI_CAUSE_FACILITY_NOTIMPL	Facility not implemented
TAPI_CAUSE_FACILITY_NOTSUBSCRIBED	Facility not subscribed
TAPI_CAUSE_INCOMINGCUGCALLS_BARRED	Incoming CUG Calls barred
TAPI_CAUSE_INVALNUM	Invalid number
TAPI_CAUSE_MPTY_ERROR	Multiparty error
TAPI_CAUSE_NOANSWER	No answer
TAPI_CAUSE_NONCUGMEMBER	Non CUG member
TAPI_CAUSE_NUMBERCHANGED	Number changed
TAPI_CAUSE_NUMBER_ERROR	Number error
TAPI_CAUSE_NWFAILURE	Network failure
TAPI_CAUSE_OPERATIONUNAVAIL	Operation not available
TAPI_CAUSE_OUTOFORDER	Out of order
TAPI_CAUSE_PROTOCOL_ERROR	Protocol error

TAPI_CAUSE_QOSUNAVAIL QOS unavailable

TAPI_CAUSE_REJECT Rejected

TAPI_CAUSE_REJ_FAIL Rejection failed

TAPI_CAUSE_REJ_SRVC_NOT_AVL Rejection service not available

TAPI_CAUSE_REMOTE_CKTUNAVAIL Remote Circuit channel unavailable

TAPI_CAUSE_RESOURCEUNAVAIL Resource not available

TAPI_CAUSE_SERVICEID_ERROR Service id error

TAPI_CAUSE_SERVICE_NOTIMPL Service not implemented

TAPI_CAUSE_SERVICE_UNAVAIL Service not available

TAPI_CAUSE_MODIFY_SAME_BEARER Modify same bearer

TAPI_CAUSE_MODIFY_NOT_PERMITTED Modification not permitted

TAPI_CAUSE_HOLD_FAIL Call hold fail

TAPI_CAUSE_TEMPFAILURE Temporary failure

TAPI_CAUSE_WRONGNUM Wrong number

TAPI_CAUSE_NORMAL_UNSPECIFIED Normal but unspecified

TAPI_CAUSE_SSERRINVALIDTIVALUE SS invalid Transaction ID Value

TAPI_CAUSE_SSERRINVALIDTI SS Invalid transaction ID

TAPI_CAUSE_SSINCOMPATIBLESTATE SS incompatible state

TAPI_CAUSE_SSERRSYNTAXERROR SS syntax error

TAPI_CAUSE_SSPROTOCOLERROR Protocol error

TAPI_CAUSE_SSNEGATIVEPASSWORDCHECK Negative password check

TAPI_CAUSE_SSSENTASPHASE1 SS sent as phase1 message

TAPI_CAUSE_SSERROR Supplementary service error

TAPI_CAUSE_SS_USSD_BUSY Second USSD request will be rejected when already USSD transaction is ongoing.

TAPI_CAUSE_IDLE Idle

TAPI_CAUSE_NETWORK_SPECIFIC Network specific error

TAPI_CAUSE_FADE voice call was dropped e.g. because of a loss of signal

TAPI_CAUSE_UNKNOWN unknown no details available

TAPI_CAUSE_INCOM_CALL terminated by incoming call

TAPI_CAUSE_ALERT_STOP terminated by alert stop

TAPI_CAUSE_INTERCEPT voice call was dropped by interception

TAPI_CAUSE_REORDER voice call was dropped by reordering

TAPI_CAUSE_CLIENT_END client ended the call

TAPI_CAUSE_ACCESS_CLASS_BLOCKED access class blocked

TAPI_CAUSE_MEMCAPEXCEEDED Memory capacity exceeded

TAPI_CAUSE_TYPENOTSUPPORTED Type not supported

TAPI_CAUSE_REPLACENOTSUPPORTED Replace not supported

TAPI_CAUSE_PROTOCOLID_ERROR Protocol id error

TAPI_CAUSE_CLASSNOTSUPPORTED Class not supported

TAPI_CAUSE_DATACODING_ERROR Data coding error

TAPI_CAUSE_INVAL_MSG Invalid message

TAPI_CAUSE_CALL_BARRED Call barred

TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_SS Sim call control changed to SS

TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_CALL SIM Call control changed to call

TAPI_CAUSE_SIM_ERROR SIM error

TAPI_CAUSE_SIM_CALL_CONTROL_MODIFIED SIM call control modified

TAPI_CAUSE_RANDOM_ACCESS_FAILURE Random access error

TAPI_CAUSE_LOWER_LAYER_FAILURE Lower layer error

TAPI_CAUSE_ACCESS_REJECTED Access rejected

TAPI_CAUSE_REEST_NOT_ALLOWED Reset not allowed

TAPI_CAUSE_EMERGENCY_CALL_NOT_ALLOWED Emergency call not allowed

TAPI_CAUSE_ACCESS_NOT_ALLOWED Access not allowed

TAPI_CAUSE_RADIO_LINK_FAILURE Radio link failure

TAPI_CAUSE_INVALID_IDENTITY Invalid identity

TAPI_CAUSE_UNKNOWN_IDENTITY Unknown identity

TAPI_CAUSE_UNACCEPTABLE_IDENTITY Un acceptable identity

TAPI_CAUSE_NO_SERVICE_HERE No service here

TAPI_CAUSE_SERVICE_NOT_ALLOWED Service not allowed

TAPI_CAUSE_SERVICE_NOT_AVAIL Service not available

TAPI_CAUSE_SERVICE_NOT_WORKING Service not working

TAPI_CAUSE_CANNOT_IDENTIFY_CALL Cannot identify the call

TAPI_CAUSE_DEACTIVATION Deactivation

TAPI_CAUSE_FATAL_ERROR Fatal error

5.91.2.5 enum TelCallCugOptions_t

This enumeration defines the closed user group options

Enumerator:

TAPI_CALL_CUG_NO_INFO No information

TAPI_CALL_CUG_SUPRESS_OA suppress OA

TAPI_CALL_CUG_SUPRESS_PRF_CUG suppress preferential CUG

TAPI_CALL_CUG_SUPRESS_OA_AND_CUG suppress OA and preferential CUG

5.91.2.6 enum TelCallIdentityMode_t

This enumeration defines the identity mode option for a call

Enumerator:

TAPI_CALL_IDENTITY_DEFAULT Calling line identity will be sent. Based on network setting, Line identity will presented to the called party.

TAPI_CALL_IDENTITY_SHOW Display calling line identity

TAPI_CALL_IDENTITY_HIDE Hide the display of Calling line identity

5.91.2.7 enum TelCallNameMode_t

This enumeration defines the call name mode

Enumerator:

TAPI_CALL_NAME_AVAIL This identifier refers to the presentation of the calling party's name identity to the called party.

TAPI_CALL_NAME_RESTRICTED This identifier refers to the restriction of presentation of the name identity of the calling party to the called party.

TAPI_CALL_NAME_UNAVAIL This identifier refers to the unavailability of the calling party's name identity to be offered to the called party.

TAPI_CALL_NAME_AVAIL_RESTRICTED This identifier refers to the calling party's name identity to be offered to the called party with which presentation restriction is overridden.

5.91.2.8 enum TelCallType_t

Applications needs to use this call type while using the setip call API. Like the requested call is a voice call or video call.

Enumerator:

TAPI_CALL_TYPE_VOICE Voice call type.

TAPI_CALL_TYPE_DATA Data call type - (for modem, fax, packet, etc. calls).

TAPI_CALL_TYPE_E911 Emergency call type.

5.91.2.9 enum TelCallUusProtocolType_t

This enumeration defines the tapi call uus protocol types

Enumerator:

TAPI_CALL_UUS_PROTO_SPECIFIC The user information is structured according to user needs.

TAPI_CALL_UUS_PROTO_OSI_HIGHER_LAYER OSI high layer protocols

TAPI_CALL_UUS_PROTO_IA5_CHARS user information consists of IA5 characters

TAPI_CALL_UUS_PROTO_V120 rate adaptation

TAPI_CALL_UUS_PROTO_Q931 user-network call control messages

5.91.2.10 enum TelCallUusType_t

This enumeration defines the UUS Type

Enumerator:

TAPI_CALL_UUS_NONE No User to User information

TAPI_CALL_UUS_1 User to User information 1

5.92 NetText Data Structures

Files

- file TelNetText.h

  This file serves as a "C" header file defines structures for Tapi Network Text Services.
  It contains a sample set of constants, enums,structs that would be required by applications.

Data Structures

- struct TelSmsCbConfig_t
- struct TelSmsAddressInfo_t
- struct TelSmsParams_t
- struct TelSmsDatapackageInfo_t
- struct TelSmsData_t
- struct TelSmsStoredMsgCountInfo_t
- struct TelSmsCbMsg_t

Defines

- #define TAPI_NETTEXT_CB_PAGE_SIZE_MAX 93
- #define TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX 90
- #define TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX 60
- #define TAPI_NETTEXT_SMDATA_SIZE_MAX 160
- #define **TAPI_NETTEXT_MAX_GSM_SMS_SERVICE_CENTER_ADDR** 12
- #define **TAPI_NETTEXT_MAX_INDEX** 255
- #define TAPI_SIM_SMSP_ADDRESS_LEN 20
- #define TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX 32

Enumerations

- enum TelSmsMsgStatus_t {

  TAPI_NETTEXT_STATUS_READ, TAPI_NETTEXT_STATUS_UNREAD, TAPI_NETTEXT_STATUS_SENT, TAPI_NETTEXT_STATUS_UNSENT,

  TAPI_NETTEXT_STATUS_DELIVERED, TAPI_NETTEXT_STATUS_DELIVERY_UNCONFIRMED, TAPI_NETTEXT_STATUS_MESSAGE_REPLACED, TAPI_NETTEXT_STATUS_RESERVED }
- enum TelSmsBearerType_t {

  TAPI_NETTEXT_BEARER_PS_ONLY = 0x01, TAPI_NETTEXT_BEARER_CS_ONLY, TAPI_NETTEXT_BEARER_PS_PREFERRED, TAPI_NETTEXT_BEARER_CS_PREFERRED,

  TAPI_NETTEXT_NO_PREFERRED_BEARER }
- enum TelSmsCbMsgType_t { TAPI_NETTEXT_CB_MSG_CBS = 1, TAPI_NETTEXT_CB_MSG_SCHEDULE, TAPI_NETTEXT_CB_MSG_CBS41, TAPI_NETTEXT_CB_MSG_INVALID }

- enum TelSmsResponse_t {

  TAPI_NETTEXT_SENDSMS_SUCCESS = 0x00, TAPI_NETTEXT_ROUTING_NOT_AVAILABLE, TAPI_NETTEXT_INVALID_PARAMETER, TAPI_NETTEXT_DEVICE_FAILURE,

  TAPI_NETTEXT_SERVICE_RESERVED, TAPI_NETTEXT_INVALID_LOCATION, TAPI_NETTEXT_NO_SIM, TAPI_NETTEXT_SIM_NOT_READY,

  TAPI_NETTEXT_NO_NETWORK_RESP, TAPI_NETTEXT_DEST_ADDRESS_FDN_RESTRICTED, TAPI_NETTEXT_SCADDRESS_FDN_RESTRICTED, TAPI_NETTEXT_RESEND_ALREADY_DONE,

  TAPI_NETTEXT_SCADDRESS_NOT_AVAILABLE, TAPI_NETTEXT_UNASSIGNED_NUMBER, TAPI_NETTEXT_OPERATOR_DETERMINED_BARRING, TAPI_NETTEXT_CALL_BARRED,

  TAPI_NETTEXT_MESSAGE_TRANSFER_REJECTED, TAPI_NETTEXT_MEMORY_CAPACITY_EXCEEDED, TAPI_NETTEXT_DESTINAITION_OUTOFSERVICE, TAPI_NETTEXT_UNSPECIFIED_SUBSCRIBER,

  TAPI_NETTEXT_FACILITY_REJECTED, TAPI_NETTEXT_UNKNOWN_SUBSCRIBER, TAPI_NETTEXT_NETWORK_OUTOFORDER, TAPI_NETTEXT_TEMPORARY_FAILURE,

  TAPI_NETTEXT_CONGESTION, TAPI_NETTEXT_RESOURCES_UNAVAILABLE, TAPI_NETTEXT_FACILITY_NOT_SUBSCRIBED, TAPI_NETTEXT_FACILITY_NOT_IMPLEMENTED,

  TAPI_NETTEXT_INVALID_REF_VALUE, TAPI_NETTEXT_INVALID_MSG, TAPI_NETTEXT_INVALID_MANDATORY_INFO, TAPI_NETTEXT_MESSAGE_TYPE_NOT_IMPLEMENTED,

  TAPI_NETTEXT_MESSAGE_NOT_COMPAT_PROTOCOL, TAPI_NETTEXT_IE_NOT_IMPLEMENTED, TAPI_NETTEXT_PROTOCOL_ERROR, TAPI_NETTEXT_INTERWORKING }
- enum TelSmsCause_t {

  TAPI_NETTEXT_SUCCESS, TAPI_NETTEXT_INVALID_PARAMETER_FORMAT, TAPI_NETTEXT_PHONE_FAILURE, TAPI_NETTEXT_SIM_BUSY,

  TAPI_NETTEXT_SIM_FAILURE, TAPI_NETTEXT_UNKNOWN, TAPI_NETTEXT_MEMORY_FAILURE, TAPI_NETTEXT_OPERATION_NOT_SUPPORTED }

5.92.1 Define Documentation

5.92.1.1 #define TAPI_NETTEXT_CB_PAGE_SIZE_MAX 93

CB maximum page size

5.92.1.2 #define TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX 60

Maximum GSM SMS CBMI list size

5.92.1.3 #define TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX 90

Maximum GSM SMS message number

5.92.1.4 #define TAPI_NETTEXT_SMDATA_SIZE_MAX 160

Maximum SMS data size that can be stored

5.92.1.5 #define TAPI_SIM_SMSP_ADDRESS_LEN 20

digit length

5.92.1.6 #define TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX 32

alpha id length

5.92.2 Enumeration Type Documentation

5.92.2.1 enum TelSmsBearerType_t

This enum defines the different bearer types

Enumerator:

TAPI_NETTEXT_BEARER_PS_ONLY Send SMS only on PS Bearer

TAPI_NETTEXT_BEARER_CS_ONLY Send SMS only on CS Bearer

TAPI_NETTEXT_BEARER_PS_PREFERRED Send SMS preferably on PS Bearer

TAPI_NETTEXT_BEARER_CS_PREFERRED Send SMS preferably on CS Bearer

TAPI_NETTEXT_NO_PREFERRED_BEARER SMS is sent based on default preferred bearer set at OEM based on vendor/operator preference

5.92.2.2 enum TelSmsCause_t

This enum defines the differnet cause types that come in the sent status acknowledgement/notification after sending a message to the network

Enumerator:

TAPI_NETTEXT_SUCCESS Sms Operation successful

TAPI_NETTEXT_INVALID_PARAMETER_FORMAT Invalid format for some parameters passed in Datapackage information(TPDU)

TAPI_NETTEXT_PHONE_FAILURE SMS operation failed due to Modem failure

TAPI_NETTEXT_SIM_BUSY SMS SIM operation cannot be performed as SIM is busy with some other operation

TAPI_NETTEXT_SIM_FAILURE SMS SIM operation cannot be performed due to sim failure

TAPI_NETTEXT_UNKNOWN unknown error

TAPI_NETTEXT_MEMORY_FAILURE Error while accessing memory or allocation of memory for SMS operation.

TAPI_NETTEXT_OPERATION_NOT_SUPPORTED operation not allowed/supported

5.92.2.3 enum TelSmsCbMsgType_t

This enumeration defines the different CB message types.

Enumerator:

TAPI_NETTEXT_CB_MSG_CBS Cell broadcast message

TAPI_NETTEXT_CB_MSG_SCHEDULE CB Schedule message

TAPI_NETTEXT_CB_MSG_CBS41 CBS41 type message

TAPI_NETTEXT_CB_MSG_INVALID Invalid CB message

5.92.2.4 enum TelSmsMsgStatus_t

This enumeration defines the network text status type.

Enumerator:

TAPI_NETTEXT_STATUS_READ MT message, stored and read

TAPI_NETTEXT_STATUS_UNREAD MT message, stored and unread

TAPI_NETTEXT_STATUS_SENT MO message, stored and sent

TAPI_NETTEXT_STATUS_UNSENT MO message, stored but not sent

TAPI_NETTEXT_STATUS_DELIVERED delivered destination

TAPI_NETTEXT_STATUS_DELIVERY_UNCONFIRMED Service centre forwarded message but is unable to confirm delivery

TAPI_NETTEXT_STATUS_MESSAGE_REPLACED Message has been replaced

TAPI_NETTEXT_STATUS_RESERVED Reserved for future use

5.92.2.5 enum TelSmsResponse_t

This enum defines the different response types that come in the sent status acknowledgement/notification after sending a message to the network

Enumerator:

TAPI_NETTEXT_SENDSMS_SUCCESS Message send success

TAPI_NETTEXT_ROUTING_NOT_AVAILABLE Message routing not available

TAPI_NETTEXT_INVALID_PARAMETER Invalid parameter present in TPDU

TAPI_NETTEXT_DEVICE_FAILURE Device failure

TAPI_NETTEXT_SERVICE_RESERVED Reserved Service

TAPI_NETTEXT_INVALID_LOCATION Invalid location

TAPI_NETTEXT_NO_SIM No sim error

TAPI_NETTEXT_SIM_NOT_READY Sim not ready error

TAPI_NETTEXT_NO_NETWORK_RESP No response from network

TAPI_NETTEXT_DEST_ADDRESS_FDN_RESTRICTED Destination address restricted

TAPI_NETTEXT_SCADDRESS_FDN_RESTRICTED Service center address restricted

TAPI_NETTEXT_RESEND_ALREADY_DONE Resend operation already done

TAPI_NETTEXT_SCADDRESS_NOT_AVAILABLE SCA address not available

TAPI_NETTEXT_UNASSIGNED_NUMBER Unassigned number

TAPI_NETTEXT_OPERATOR_DETERMINED_BARRING Operator determined barring

TAPI_NETTEXT_CALL_BARRED Call barred

TAPI_NETTEXT_MESSAGE_TRANSFER_REJECTED Message transfer rejected

TAPI_NETTEXT_MEMORY_CAPACITY_EXCEEDED Memory capacity exceeded/memory full

TAPI_NETTEXT_DESTINAITION_OUTOFSERVICE Destination number/address out of service

TAPI_NETTEXT_UNSPECIFIED_SUBSCRIBER Unspecified subscriber

TAPI_NETTEXT_FACILITY_REJECTED Facility rejected

TAPI_NETTEXT_UNKNOWN_SUBSCRIBER Unknown subscriber

TAPI_NETTEXT_NETWORK_OUTOFORDER Network out of order

TAPI_NETTEXT_TEMPORARY_FAILURE Temporary failure

TAPI_NETTEXT_CONGESTION Congestion happened

TAPI_NETTEXT_RESOURCES_UNAVAILABLE Resource unavailable

TAPI_NETTEXT_FACILITY_NOT_SUBSCRIBED Facility not subscribed by the user

TAPI_NETTEXT_FACILITY_NOT_IMPLEMENTED Facility not implemented

TAPI_NETTEXT_INVALID_REF_VALUE Invalid reference value

TAPI_NETTEXT_INVALID_MSG Invalid message

TAPI_NETTEXT_INVALID_MANDATORY_INFO Invalid mandatory information

TAPI_NETTEXT_MESSAGE_TYPE_NOT_IMPLEMENTED Message type not implemented

TAPI_NETTEXT_MESSAGE_NOT_COMPAT_PROTOCOL Message not compact protocol

TAPI_NETTEXT_IE_NOT_IMPLEMENTED Information element not implemented

TAPI_NETTEXT_PROTOCOL_ERROR Protocol error

TAPI_NETTEXT_INTERWORKING Networking error

5.93 Network Data Structures

Files

- file TelNetwork.h

  This file serves as a "C" header file defines structures for Tapi Network Services.
  It contains a sample set of constants, enums,structs that would be required by applications.

Data Structures

- struct TelNetworkTimeAndZone_t
- struct TelNetworkSystemId_t
- struct TelNetworkConfigSettings_t
- struct TelNetworkInfo_t
- struct TelNetworkList_t
- struct TelNetworkRadioInfo_t

Defines

- #define TAPI_NETWORK_ORDER_MAX 5
- #define TAPI_NETWORK_LIST_MAX 15
- #define TAPI_NETWORK_NAME_LEN_MAX 40
- #define TAPI_NETWORK_SPN_LEN_MAX 20

Enumerations

- enum TelNetworkSelectionMode_t { TAPI_NETWORK_SELECTIONMODE_AUTOMATIC, TAPI_NETWORK_SELECTIONMODE_MANUAL }
- enum TelNetworkSystemIdFormat_t { TAPI_NETWORK_SYSID_FORMAT_SID, TAPI_NETWORK_SYSID_FORMAT_PLMN }
- enum TelNetworkServiceDomain_t { TAPI_NETWORK_SERVICE_DOMAIN_COMBINED, TAPI_NETWORK_SERVICE_DOMAIN_PS, TAPI_NETWORK_SERVICE_DOMAIN_CS, TAPI_NETWORK_SERVICE_DOMAIN_AUTOMATIC }
- enum TelNetworkMode_t {

  TAPI_NETWORK_MODE_EMERGENCY, TAPI_NETWORK_MODE_AUTOMATIC, TAPI_NETWORK_MODE_CDMA, TAPI_NETWORK_MODE_GSM,

  TAPI_NETWORK_MODE_WCDMA, TAPI_NETWORK_MODE_WLAN }
- enum TelNetworkServiceLevel_t { TAPI_NETWORK_SERVICE_LEVEL_NO, TAPI_NETWORK_SERVICE_LEVEL_EMERGENCY, TAPI_NETWORK_SERVICE_LEVEL_FULL, TAPI_NETWORK_SERVICE_LEVEL_SEARCH }
- enum TelNetworkServiceType_t {

  TAPI_NETWORK_SERVICE_TYPE_NO_SVE, TAPI_NETWORK_SERVICE_TYPE_LIMITED, TAPI_NETWORK_SERVICE_TYPE_CS, TAPI_NETWORK_SERVICE_TYPE_PS,

  TAPI_NETWORK_SERVICE_TYPE_COMBINED }

- enum TelNetworkSystemType_t {

  TAPI_NETWORK_SYSTEM_NO_SRV, TAPI_NETWORK_SYSTEM_CDMA, TAPI_NETWORK_SYSTEM_TDMA, TAPI_NETWORK_SYSTEM_GSM,

  TAPI_NETWORK_SYSTEM_GPRS, TAPI_NETWORK_SYSTEM_EGPRS, TAPI_NETWORK_SYSTEM_PCS1900, TAPI_NETWORK_SYSTEM_UMTS,

  TAPI_NETWORK_SYSTEM_GSM_AND_UMTS }
- enum TelNetworkBandPreferred_t { TAPI_NETWORK_BAND_MODE_PREFERRED = 0x01, TAPI_NETWORK_BAND_MODE_ONLY = 0x02 }
- enum TelNetworkBand_t {

  TAPI_NETWORK_BAND_TYPE_ANY = 0x00, TAPI_NETWORK_BAND_TYPE_GSM850, TAPI_NETWORK_BAND_TYPE_GSM_900_1800, TAPI_NETWORK_BAND_TYPE_GSM1900,

  TAPI_NETWORK_BAND_TYPE_GSM, TAPI_NETWORK_BAND_TYPE_WCDMA, TAPI_NETWORK_BAND_TYPE_WCDMA850, TAPI_NETWORK_BAND_TYPE_WCDMA1900,

  TAPI_NETWORK_BAND_TYPE_WCDMA2100 }
- enum TelNetworkOperationCause_t {

  TAPI_NETWORK_NO_ERR = 0x00, TAPI_NETWORK_OPERATION_FAILED, TAPI_NETWORK_PHONE_IN_USE_ERR, TAPI_NETWORK_OFFLINE_ERR,

  TAPI_NETWORK_CONFIG_SETTINGS_FAILURE_ERR, TAPI_NETWORK_INTERNAL_FAILURE_ERR }
- enum TelNetworkDayLightSavingAdj_t { TAPI_NETWORK_DLS_ADJ_NO_ADJUSTMENT, TAPI_NETWORK_DLS_ADJ_PLUS_ONE_HOUR, TAPI_NETWORK_DLS_PLUS_TWO_HOUR }

5.93.1 Define Documentation

5.93.1.1 #define TAPI_NETWORK_LIST_MAX 15

Maximum network list

5.93.1.2 #define TAPI_NETWORK_NAME_LEN_MAX 40

Maximum network name or PLMN name length

5.93.1.3 #define TAPI_NETWORK_ORDER_MAX 5

Maximum network order

5.93.1.4 #define TAPI_NETWORK_SPN_LEN_MAX 20

Maximum network service provider name length

5.93.2 Enumeration Type Documentation

5.93.2.1 enum TelNetworkBand_t

This enumeration defines different network Bands.

Enumerator:

- ***TAPI_NETWORK_BAND_TYPE_ANY*** All bands can be tried.
- ***TAPI_NETWORK_BAND_TYPE_GSM850*** This member sets the band to GSM 850
- ***TAPI_NETWORK_BAND_TYPE_GSM_900_1800*** This member sets the band to GSM 900_1800
- ***TAPI_NETWORK_BAND_TYPE_GSM1900*** GSM 1900
- ***TAPI_NETWORK_BAND_TYPE_GSM*** This member sets all GSM bands
- ***TAPI_NETWORK_BAND_TYPE_WCDMA*** This member sets all WCDMA bands
- ***TAPI_NETWORK_BAND_TYPE_WCDMA850*** This member sets the band to WCDMA 850
- ***TAPI_NETWORK_BAND_TYPE_WCDMA1900*** This member sets the band to WCDMA 1900
- ***TAPI_NETWORK_BAND_TYPE_WCDMA2100*** This member sets the band to WCDMA 2100

5.93.2.2 enum TelNetworkBandPreferred_t

This enumeration defines the opeartions that can be done on network Band.

Enumerator:

- ***TAPI_NETWORK_BAND_MODE_PREFERRED*** Preferred band. (If preferred Band is not available, then underlying modem can search on other bands)
- ***TAPI_NETWORK_BAND_MODE_ONLY*** This allows only particular band.

5.93.2.3 enum TelNetworkDayLightSavingAdj_t

This enumeration defines the day light saving adjustment time.

Enumerator:

- ***TAPI_NETWORK_DLS_ADJ_NO_ADJUSTMENT*** No adjustment specified on day light saving
- ***TAPI_NETWORK_DLS_ADJ_PLUS_ONE_HOUR*** Day light saving adjusted for one hour
- ***TAPI_NETWORK_DLS_PLUS_TWO_HOUR*** Day light saving adjusted for two hours

5.93.2.4 enum TelNetworkMode_t

This enumeration defines the network modes.

Enumerator:

- ***TAPI_NETWORK_MODE_EMERGENCY*** This member sets the emergency network mode
- ***TAPI_NETWORK_MODE_AUTOMATIC*** This member sets the automatic network mode. Based on OEM priority, nw mode preference will be used
- ***TAPI_NETWORK_MODE_CDMA*** This member sets the network mode to CDMA
- ***TAPI_NETWORK_MODE_GSM*** This member sets the network mode to GSM
- ***TAPI_NETWORK_MODE_WCDMA*** This member sets the network mode to WCDMA
- ***TAPI_NETWORK_MODE_WLAN*** This member sets the network mode to WLAN

5.93.2.5 enum TelNetworkOperationCause_t

This enumeration defines different network operation cause.

Enumerator:

TAPI_NETWORK_NO_ERR There is no error for any network operation.

TAPI_NETWORK_OPERATION_FAILED There is no error for any network operation.

TAPI_NETWORK_PHONE_IN_USE_ERR This error will be returned when any network operation is tried while phone is in use(eg, Voice / Data call in progress due to which network operation failed)

TAPI_NETWORK_OFFLINE_ERR This error is sent when any network option is tried when phone is in offline mode

TAPI_NETWORK_CONFIG_SETTINGS_FAILURE_ERR This error is sent when modem is unable to process the config settings information.

TAPI_NETWORK_INTERNAL_FAILURE_ERR This error to indicate internal failure

5.93.2.6 enum TelNetworkSelectionMode_t

This enumeration defines the network selection modes.

Enumerator:

TAPI_NETWORK_SELECTIONMODE_AUTOMATIC Network selection is in Automatic selection mode

TAPI_NETWORK_SELECTIONMODE_MANUAL Network selection is in Manual selection mode

5.93.2.7 enum TelNetworkServiceDomain_t

This enumeration defines the network service domains .

Enumerator:

TAPI_NETWORK_SERVICE_DOMAIN_COMBINED This member sets the Circuit switch and Packet Switch domain

TAPI_NETWORK_SERVICE_DOMAIN_PS This member sets the Packet Switch service domain

TAPI_NETWORK_SERVICE_DOMAIN_CS This member sets the Circuit Switch service domain

TAPI_NETWORK_SERVICE_DOMAIN_AUTOMATIC This member sets the service domain to Circuit Switch and Packet Switch or Circuit Switch or Packet Switch domain

5.93.2.8 enum TelNetworkServiceLevel_t

This enumeration defines the status of network service.

Enumerator:

TAPI_NETWORK_SERVICE_LEVEL_NO No service available in the network ME is camped

TAPI_NETWORK_SERVICE_LEVEL_EMERGENCY Only emergency service available in the network ME is camped

TAPI_NETWORK_SERVICE_LEVEL_FULL FULL service available in the network ME is camped

TAPI_NETWORK_SERVICE_LEVEL_SEARCH Searching for service

5.93.2.9 enum TelNetworkServiceType_t

This enumeration defines the service types of TAPI.

Enumerator:

TAPI_NETWORK_SERVICE_TYPE_NO_SVE No service (This enumeration is not applicable for OEM_TAPI_NW_select_network_req API)

TAPI_NETWORK_SERVICE_TYPE_LIMITED Limited service available in the camped network (This enumeration is not applicable for OEM_TAPI_NW_select_network_req API)

TAPI_NETWORK_SERVICE_TYPE_CS CS service available in the camped network

TAPI_NETWORK_SERVICE_TYPE_PS PS service available in the camped network

TAPI_NETWORK_SERVICE_TYPE_COMBINED Both CS and PS services available in the camped network

5.93.2.10 enum TelNetworkSystemIdFormat_t

This enumeration defines the System ID Formats.

Enumerator:

TAPI_NETWORK_SYSID_FORMAT_SID System ID in CDMA.

TAPI_NETWORK_SYSID_FORMAT_PLMN PLMN id in GSM/3G

5.93.2.11 enum TelNetworkSystemType_t

This enumeration defines the system types of TAPI.

Enumerator:

TAPI_NETWORK_SYSTEM_NO_SRV No Service available

TAPI_NETWORK_SYSTEM_CDMA Available service is CDMA

TAPI_NETWORK_SYSTEM_TDMA Available service is TDMA

TAPI_NETWORK_SYSTEM_GSM Available service is GSM

TAPI_NETWORK_SYSTEM_GPRS Available service is GPRS

TAPI_NETWORK_SYSTEM_EGPRS Available service is EGPRS

TAPI_NETWORK_SYSTEM_PCS1900 Available service is PCS1900 band

TAPI_NETWORK_SYSTEM_UMTS Available service is UMTS

TAPI_NETWORK_SYSTEM_GSM_AND_UMTS Both GSM and UMTS systems available

5.94 SIM Data Structures

Files

- file TelSim.h

  This file serves as a "C" header file defines structures for Tapi SIM Services.
  It contains a sample set of constants, enums,structs that would be required by applications.

Data Structures

- struct TelSimFileID_t
- struct TelSimPinData_t
- struct TelSimReadFileRequestData_t
- struct TelSimIccIdInfo_t
- struct TelSimImsiInfo_t
- struct TelSimServiceTable_t
- struct TelSimSstInfo_t
- struct TelSimPinInfo_t
- struct TelSimPb2gData_t
- struct TelSimPbRecordData_t
- struct TelSimEccInfo_t
- struct TelUsimEccInfo_t
- struct TelUSimServiceTable_t
- struct TelSimUstInfo_t
- struct TelSimPbStorageInfo_t
- struct TelSimPbEntryInfo_t
- struct TelSimPbReadResp_t
- struct TelSimPbUpdateResp_t
- struct TelSimLanguageInfo_t
- struct TelSimFileAccessGeneralResp_t
- struct TelSimGroupIdentifierInfo_t
- struct TelSimEccData_t
- struct TelSimStatusData_t
- struct TelSimCallbackEvent_t
- struct TelSimEvent_t
- struct TelSimApnData_t
- struct TelSimDeviceContextData_t
- struct TelSimDeviceContext_t
- struct TelSimFileInfo_t
- struct TelSimReadFileData_t
- struct TelSimUpdateFileRequestData_t
- struct TelSimFileInfo_t

Defines

- #define TAPI_SIM_PIN_LENGTH 8
- #define TAPI_SIM_ICCIDLENGTH 10
- #define TAPI_SIM_ICCID_PACKET_LEN 10
- #define TAPI_SIM_ICCID_MAX_LEN 10
- #define TAPI_SIM_LP_MAX_LEN 30
- #define TAPI_SIM_LP_PACKET_MAX_LEN 30
- #define TAPI_SIM_IMSI_LEN 8
- #define TAPI_SIM_IMSI_PACKET_LEN 15
- #define TAPI_SIM_MCC_CODE_LEN 3
- #define TAPI_SIM_MNC_CODE_LEN 2
- #define TAPI_SIM_MSIN_CODE_LEN 10
- #define TAPI_SIM_NUM_TEXT_MAX_LEN 256
- #define TAPI_SIM_SST_MAX_LEN 20
- #define TAPI_SIM_GSM_SST_SERVICE_MAX_COUNT 50
- #define TAPI_SIM_ECC_CODE_LEN 3
- #define TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX 50
- #define TAPI_TAPI_SIM_PBOOK_ENTRY_COUNT_MAX 20
- #define TAPI_SIM_LI_MAX_LEN 30
- #define TAPI_SIM_LI_PACKET_MAX_LEN 30
- #define TAPI_SIM_ACL_MAX_LEN 50
- #define TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX 10
- #define TAPI_SIM_ECC_MAX_LEN 15
- #define TAPI_SIM_UECC_RECORD_COUNT_MAX 5
- #define TAPI_SIM_UECC_RECORD_LEN_MAX 50

Enumerations

- enum TelSimCardType_t { TAPI_SIM_CARD_TYPE_UNKNOWN, TAPI_SIM_CARD_TYPE_GSM, TAPI_SIM_CARD_TYPE_USIM }
- enum TelSimFileName_t {

  TAPI_SIM_EFILE_DIR = 0x2F00, TAPI_SIM_EFILE_ICCID = 0x2FE2, TAPI_SIM_EFILE_ADN = 0x6F3A, TAPI_SIM_EFILE_FDN = 0x6F3B,

  TAPI_SIM_EFILE_SMS = 0x6F3C, TAPI_SIM_EFILE_CCP = 0x6F3D, TAPI_SIM_EFILE_IMSI = 0x6F07, TAPI_SIM_EFILE_SST = 0x6F38,

  TAPI_SIM_EFILE_EST = 0x6F56, TAPI_SIM_EFILE_ACM = 0x6F39, TAPI_SIM_EFILE_MBDN = 0x6FC7, TAPI_SIM_EFILE_GID1 = 0x6F3E,

  TAPI_SIM_EFILE_GID2 = 0x6F3F, TAPI_SIM_EFILE_MSISDN = 0x6F40, TAPI_SIM_EFILE_SMSP = 0x6F42, TAPI_SIM_EFILE_SMSS = 0x6F43,

  TAPI_SIM_EFILE_EXT1 = 0x6F4A, TAPI_SIM_EFILE_EXT2 = 0x6F4B, TAPI_SIM_EFILE_LP = 0x2F05, TAPI_SIM_EFILE_ACM_MAX = 0x6F37,

  TAPI_SIM_EFILE_PUCT = 0x6F41, TAPI_SIM_EFILE_CBMI = 0x6F45, TAPI_SIM_EFILE_PHASE = 0x6FAE, TAPI_SIM_EFILE_SDN = 0x6F49,

  TAPI_SIM_EFILE_EXT3 = 0x6F4C, TAPI_SIM_EFILE_BDN = 0x6F4D, TAPI_SIM_EFILE_EXT4 = 0x6F4E, TAPI_SIM_EFILE_ECC = 0x6FB7,

  TAPI_SIM_EFILE_SPN = 0x6F46, TAPI_SIM_EFILE_PNN = 0x6FC5, TAPI_SIM_EFILE_IMG = 0x4F20, TAPI_SIM_EFILE_USIM_LI = 0x6F05, TAPI_SIM_EFILE_USIM_IMSI = 0x6A07, TAPI_SIM_EFILE_USIM_ACM_MAX = 0x6A37, TAPI_SIM_EFILE_USIM_UST = 0x6A38, TAPI_SIM_EFILE_USIM_ACM = 0x6A39,

TAPI_SIM_EFILE_USIM_SPN = 0x6A46, TAPI_SIM_EFILE_USIM_PUCT = 0x6A41, TAPI_SIM_EFILE_USIM_CBMI = 0x6A45, TAPI_SIM_EFILE_USIM_ECC = 0x6AB7,

TAPI_SIM_EFILE_USIM_FDN = 0x6A3B, TAPI_SIM_EFILE_USIM_SMS = 0x6A3C, TAPI_SIM_EFILE_USIM_MSISDN = 0x6A40, TAPI_SIM_EFILE_USIM_SMSP = 0x6A42,

TAPI_SIM_EFILE_USIM_SMSS = 0x6A43, TAPI_SIM_EFILE_USIM_SDN = 0x6A49, TAPI_SIM_EFILE_USIM_EXT1 = 0X6A4A, TAPI_SIM_EFILE_USIM_EXT2 = 0x6A4B,

TAPI_SIM_EFILE_USIM_EXT3 = 0x6A4C, TAPI_SIM_EFILE_USIM_EXT4 = 0x6A55, TAPI_SIM_EFILE_USIM_EXT5 = 0x6A4E, **TAPI_SIM_EFILE_USIM_EXT6** = 0x6FC8,

TAPI_SIM_EFILE_USIM_CCP2 = 0x6A4F, TAPI_SIM_EFILE_USIM_HIDDEN_KEY = 0x6FC3, TAPI_SIM_EFILE_USIM_BDN = 0x6A4D, TAPI_SIM_EFILE_USIM_MBDN = 0x6FC7,

TAPI_SIM_EFILE_USIM_MBI = 0x6FC9, TAPI_SIM_EFILE_USIM_PHONEBOOK_PBR = 0x4FF0, TAPI_SIM_EFILE_USIM_PHONEBOOK_CCP1 = 0x4FFE, TAPI_SIM_EFILE_USIM_PHONEBOOK_UID = 0x4FF8,

TAPI_SIM_EFILE_USIM_PHONEBOOK_PSC = 0x4FFF, TAPI_SIM_EFILE_USIM_PHONEBOOK_CC = 0x4FEF, TAPI_SIM_EFILE_USIM_PHONEBOOK_PUID = 0x4FEE, TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN = 0x4FF1,

TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP = 0x4FF2, TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC = 0x4FF3, TAPI_SIM_EFILE_USIM_PHONEBOOK_EXT1 = 0x4FFB, TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP = 0x4FF7,

TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS = 0x4FFC, TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS1 = 0x4FFD, TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS = 0x4F0E, TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS1 = 0x4F0F,

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR = 0x4FF4, TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE = 0x4FF9, TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL = 0x4FFA, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA = 0x4F13,

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB = 0x4F14, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC = 0x4F15, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR1 = 0x4F16, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA1 = 0x4F17,

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB1 = 0x4F18, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC1 = 0x4F21, TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN1 = 0x4F1A, TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC1 = 0x4F1B,

TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP1 = 0x4F1C, TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE1 = 0x4F1D, TAPI_SIM_EFILE_USIM_PHONEBOOK_UID1 = 0x4F1E, TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL1 = 0x4F1F,

TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP1 = 0x4FF5, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR = 0x4F30, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR1 = 0x4F29, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR2 = 0x4F28,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CCP1 = 0x4F3D, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID = 0x4F00, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PSC = 0x4F22, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CC = 0x4F23,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PUID = 0x4F24, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN = 0x4F3A, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP = 0x4F25, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC = 0x4F09,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EXT1 = 0x4F4A, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP = 0x4F26, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_AAS = 0x4F4B, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GAS = 0x4F4C, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR = 0x4F11, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE = 0x4F19, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL = 0x4F50, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA = 0x4F31,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB = 0x4F32, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRC = 0x4F40, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR1 = 0x4F33, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA1 = 0x4F34,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB1 = 0x4F35, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN1 = 0x4F36, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC1 = 0x4F37, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP1 = 0x4F38,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE1 = 0x4F39, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID1 = 0x4F3B, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL1 = 0x4F3C, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP1 = 0x4F3E,

TAPI_SIM_EFILE_MSG_WAITING_INDICATION_STATUS = 0x6FCA, TAPI_SIM_EFILE_INVALID = 0xFFFF, TAPI_SIM_EFILE_OTHERS }

- enum TelSimFileType_t {

  TAPI_SIM_FTYPE_DEDICATED = 0x00, TAPI_SIM_FTYPE_TRANSPARENT = 0x01, TAPI_SIM_FTYPE_LINEAR_FIXED = 0x02, TAPI_SIM_FTYPE_CYCLIC = 0x04,

  TAPI_SIM_FTYPE_INVALID_TYPE = 0xFF }

- enum TelSimRecordMode_t {

  TAPI_SIM_RECORD_NONE = 0x00, TAPI_SIM_RECORD_ABSOLUTE = 0x01, TAPI_SIM_RECORD_CURRENT = 0x02, TAPI_SIM_RECORD_NEXT = 0x03,

  TAPI_SIM_RECORD_PREVIOUS = 0x04, TAPI_SIM_RECORD_TO_THE_LAST = 0x05, TAPI_SIM_RECORD_FROM_THE_FIRST = 0x06 }

- enum TelSimPinStatus_t {

  TAPI_SIM_PIN_STATUS_DISABLED = 0x00, TAPI_SIM_PIN_STATUS_ENABLED = 0x01, TAPI_SIM_PIN_STATUS_BLOCKED = 0x02, TAPI_SIM_PIN_STATUS_PUK_BLOCKED = 0x03,

  TAPI_SIM_PIN_STATUS_UNKNOWN = 0xFF }

- enum TelSimPinOperationResult_t {

  TAPI_SIM_PIN_OPERATION_SUCCESS, TAPI_SIM_BUSY_STATUS, TAPI_SIM_CARD_ERROR_STATUS, TAPI_SIM_INCOMPATIBLE_PIN_OPERATION_STATUS,

  TAPI_SIM_PIN_INCORRECT_PASSWORD_STATUS, TAPI_SIM_PIN2_REQD_STATUS, TAPI_SIM_PIN2_INCORRECT_PASSWORD_STATUS, TAPI_SIM_PUK_INCORRECT_PASSWORD_STATUS,

  TAPI_SIM_PUK_REQD_STATUS, TAPI_SIM_PUK2_REQD_STATUS }

- enum TelSimAccessResult_t {

  TAPI_SIM_ACCESS_SUCCESS, TAPI_SIM_ACCESS_CARD_ERROR, TAPI_SIM_ACCESS_FILE_NOT_FOUND, TAPI_SIM_ACCESS_ACCESS_CONDITION_NOT_SATISFIED,

  TAPI_SIM_ACCESS_FAILED }

- enum TelSimPinType_t {

  TAPI_SIM_PTYPE_PIN1 = 0x00, TAPI_SIM_PTYPE_PIN2 = 0x01, TAPI_SIM_PTYPE_PUK1 = 0x02, TAPI_SIM_PTYPE_PUK2 = 0x03,

  TAPI_SIM_PTYPE_UPIN = 0x04, TAPI_SIM_PTYPE_ADM = 0x05 }

- enum TelSimTypeOfNum_t {

  TAPI_SIM_TON_UNKNOWN = 0, TAPI_SIM_TON_INTERNATIONAL = 1, TAPI_SIM_TON_NATIONAL = 2, TAPI_SIM_TON_NETWORK_SPECIFIC = 3,

  TAPI_SIM_TON_DEDICATED_ACCESS = 4, TAPI_SIM_TON_ALPHA_NUMERIC = 5, TAPI_SIM_TON_ABBREVIATED_NUMBER = 6, TAPI_SIM_TON_RESERVED_FOR_EXT = 7 }

- enum TelSimTextEncrypt_t { TAPI_SIM_TEXT_ENC_ASCII, TAPI_SIM_TEXT_ENC_GSM7BIT, TAPI_SIM_TEXT_ENC_UCS2, TAPI_SIM_TEXT_ENC_HEX }
- enum TelSimPbFileType_t {

  TAPI_SIM_PB_FDN, TAPI_SIM_PB_LDN, TAPI_SIM_PB_MSISDN, TAPI_SIM_PB_ADN,

  TAPI_SIM_PB_SDN }
- enum TelSimPbAccessMode_t { TAPI_SIM_PB_ACCESS_MODE_ADD = 0x01, TAPI_SIM_PB_ACCESS_MODE_DEL = 0x02, TAPI_SIM_PB_ACCESS_MODE_EDIT = 0x03 }
- enum TelSimNumberingPlanIdentity_t {

  TAPI_SIM_NPI_UNKNOWN = 0, TAPI_SIM_NPI_ISDN_TEL = 1, TAPI_SIM_NPI_DATA_NUMBERING_PLAN = 3, TAPI_SIM_NPI_TELEX = 4,

  TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN = 5, TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN2 = 6, TAPI_SIM_NPI_NATIONAL = 8, TAPI_SIM_NPI_PRIVATE = 9,

  TAPI_SIM_NPI_ERMES_NUMBERING_PLAN = 10, TAPI_SIM_NPI_RESERVED_FOR_EXT = 0xF }
- enum TelSimEccEmergencyServiceInfo_t {

  TAPI_SIM_ECC_ESC_POLICE = 0x01, TAPI_SIM_ECC_ESC_AMBULANCE = 0x02, TAPI_SIM_ECC_ESC_FIREBRIGADE = 0x03, TAPI_SIM_ECC_ESC_MARAINEGUARD = 0x04,

  TAPI_SIM_ECC_ESC_MOUTAINRESCUE = 0x05, TAPI_SIM_ECC_ESC_SPARE = 0x00 }
- enum TelSimPbAccessResult_t {

  TAPI_SIM_PB_SUCCESS, TAPI_SIM_PB_FAIL, TAPI_SIM_PB_INVALID_INDEX, TAPI_SIM_PB_INVALID_NUMBER_LENGTH,

  TAPI_SIM_PB_INVALID_NAME_LENGTH, TAPI_SIM_PB_ACCESS_CONDITION_NOT_SATISFIED }
- enum TelSimLangIndicationCode_t {

  TAPI_SIM_LI_GERMAN = 0, TAPI_SIM_LI_ENGLISH = 0x01, TAPI_SIM_LI_ITALIAN = 0x02, TAPI_SIM_LI_FRENCH = 0x03,

  TAPI_SIM_LI_SPANISH = 0x04, TAPI_SIM_LI_DUTCH = 0x05, TAPI_SIM_LI_SWEDISH = 0x06, TAPI_SIM_LI_DANISH = 0x07,

  TAPI_SIM_LI_PORTUGUESE = 0x08, TAPI_SIM_LI_FINNISH = 0x09, TAPI_SIM_LI_NORWEGIAN = 0x0A, TAPI_SIM_LI_GREEK = 0x0B,

  TAPI_SIM_LI_TURKISH = 0x0C, TAPI_SIM_LI_HUNGARIAN = 0x0D, TAPI_SIM_LI_POLISH = 0x0E, TAPI_SIM_LI_LANG_UNSPECIFIED = 0x0F }
- enum TelSimAccessType_t {

  TAPI_SIM_SELECT, TAPI_SIM_READ_BINARY, TAPI_SIM_READ_RECORD, TAPI_SIM_GET_RESPONSE,

  TAPI_SIM_UPDATE_BINARY, TAPI_SIM_UPDATE_RECORD, TAPI_SIM_STATUS }
- enum TelSimCardStatus_t {

  TAPI_SIM_STATUS_CARD_ERROR = 0x00, TAPI_SIM_STATUS_CARD_NOT_PRESENT = 0x01, TAPI_SIM_STATUS_SIM_INITIALIZING = 0x02, TAPI_SIM_STATUS_SIM_INIT_COMPLETED = 0x03,

  TAPI_SIM_STATUS_SIM_PIN_REQUIRED = 0x04, TAPI_SIM_STATUS_SIM_PUK_REQUIRED = 0x05, TAPI_SIM_STATUS_CARD_BLOCKED = 0x06 }
- enum TelSimAcl_t {

  TAPI_SIM_ACL_ALW = 0x00, TAPI_SIM_ACL_PIN1 = 0x01, TAPI_SIM_ACL_PIN2 = 0x02, TAPI_SIM_ACL_RFU = 0x04,

  TAPI_SIM_ACL_ADM = 0x08, TAPI_SIM_ACL_NEVER = 0xFF }

5.94.1 Define Documentation

5.94.1.1 #define TAPI_SIM_ACL_MAX_LEN 50

Maximum ACL length

5.94.1.2 #define TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX 50

Maximum ECC Alpha string length

5.94.1.3 #define TAPI_SIM_ECC_CODE_LEN 3

Maximum ECC code length

5.94.1.4 #define TAPI_SIM_ECC_MAX_LEN 15

Maximum ECC length

5.94.1.5 #define TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX 10

Maximum group identifier length

5.94.1.6 #define TAPI_SIM_GSM_SST_SERVICE_MAX_COUNT 50

Maximum SST service count

5.94.1.7 #define TAPI_SIM_ICCID_MAX_LEN 10

Maximum length of ICCID

5.94.1.8 #define TAPI_SIM_ICCID_PACKET_LEN 10

ICCID Packet Length

5.94.1.9 #define TAPI_SIM_ICCIDLENGTH 10

ICCID Length

5.94.1.10 #define TAPI_SIM_IMSI_LEN 8

IMSI length

5.94.1.11 #define TAPI_SIM_IMSI_PACKET_LEN 15

maximum IMSI packet length

5.94.1.12 #define TAPI_SIM_LI_MAX_LEN 30

Maximum Language indication length

5.94.1.13 #define TAPI_SIM_LI_PACKET_MAX_LEN 30

maximum LI Packet length

5.94.1.14 #define TAPI_SIM_LP_MAX_LEN 30

Language preference maximum length

5.94.1.15 #define TAPI_SIM_LP_PACKET_MAX_LEN 30

Language preference packet maximum length

5.94.1.16 #define TAPI_SIM_MCC_CODE_LEN 3

maximum MCC code length

5.94.1.17 #define TAPI_SIM_MNC_CODE_LEN 2

maximum MNC Code length

5.94.1.18 #define TAPI_SIM_MSIN_CODE_LEN 10

Maximum MSIN code length

5.94.1.19 #define TAPI_SIM_NUM_TEXT_MAX_LEN 256

Maximum Text length for number

5.94.1.20 #define TAPI_SIM_PIN_LENGTH 8

max length for PIN

5.94.1.21 #define TAPI_SIM_SST_MAX_LEN 20

Maximum SST code length

5.94.1.22 #define TAPI_SIM_UECC_RECORD_COUNT_MAX 5

Maximum UECC Record count

5.94.1.23 #define TAPI_SIM_UECC_RECORD_LEN_MAX 50

Maximum UECC record length

5.94.1.24 #define TAPI_TAPI_SIM_PBOOK_ENTRY_COUNT_MAX 20

max 254 entries within a record of PBR

5.94.2 Enumeration Type Documentation

5.94.2.1 enum TelSimAccessResult_t

This enumeration defines the SIM access result from the lower layers.

Enumerator:

TAPI_SIM_ACCESS_SUCCESS Access to file successful.

TAPI_SIM_ACCESS_CARD_ERROR Sim card error

TAPI_SIM_ACCESS_FILE_NOT_FOUND File not found

TAPI_SIM_ACCESS_ACCESS_CONDITION_NOT_SATISFIED Access condition is not fulfilled

TAPI_SIM_ACCESS_FAILED Access failed.

5.94.2.2 enum TelSimAccessType_t

This enumeration defines the access type information.

Enumerator:

TAPI_SIM_SELECT Select SIM

TAPI_SIM_READ_BINARY Read binary file

TAPI_SIM_READ_RECORD Read record

TAPI_SIM_GET_RESPONSE Get response

TAPI_SIM_UPDATE_BINARY Update binary

TAPI_SIM_UPDATE_RECORD Update record

TAPI_SIM_STATUS SIM status

5.94.2.3 enum TelSimAcl_t

This enumeration defines the ACL.

Enumerator:

TAPI_SIM_ACL_ALW ALWays

TAPI_SIM_ACL_PIN1 PIN 1

TAPI_SIM_ACL_PIN2 PIN 2

TAPI_SIM_ACL_RFU RFU

TAPI_SIM_ACL_ADM Administration Key

TAPI_SIM_ACL_NEVER Never

5.94.2.4 enum TelSimCardStatus_t

This enumeration defines the SIM card status

Enumerator:

- ***TAPI_SIM_STATUS_CARD_ERROR*** Bad card / On the fly SIM gone bad
- ***TAPI_SIM_STATUS_CARD_NOT_PRESENT*** Card not present / removed
- ***TAPI_SIM_STATUS_SIM_INITIALIZING*** Sim is Initializing state
- ***TAPI_SIM_STATUS_SIM_INIT_COMPLETED*** Sim Initialization ok
- ***TAPI_SIM_STATUS_SIM_PIN_REQUIRED*** PIN required state
- ***TAPI_SIM_STATUS_SIM_PUK_REQUIRED*** PUK required state
- ***TAPI_SIM_STATUS_CARD_BLOCKED*** PIN/PUK blocked(permanently blocked- All the attempts for PIN/PUK failed)

5.94.2.5 enum TelSimCardType_t

This enumeration defines the card type.

Enumerator:

- ***TAPI_SIM_CARD_TYPE_UNKNOWN*** Unkown card
- ***TAPI_SIM_CARD_TYPE_GSM*** Gsm card
- ***TAPI_SIM_CARD_TYPE_USIM*** Usim card

5.94.2.6 enum TelSimEccEmergencyServiceInfo_t

This enumeration defines the emergency service type.

Enumerator:

- ***TAPI_SIM_ECC_ESC_POLICE*** Police
- ***TAPI_SIM_ECC_ESC_AMBULANCE*** Ambulance
- ***TAPI_SIM_ECC_ESC_FIREBRIGADE*** Fire brigade
- ***TAPI_SIM_ECC_ESC_MARAINEGUARD*** Marine guard
- ***TAPI_SIM_ECC_ESC_MOUTAINRESCUE*** Mountain rescue
- ***TAPI_SIM_ECC_ESC_SPARE*** Spare

5.94.2.7 enum TelSimFileName_t

This enumeration defines the file id.

Enumerator:

- ***TAPI_SIM_EFILE_DIR*** for the USIM
- ***TAPI_SIM_EFILE_ICCID*** the ICC Identification file
- ***TAPI_SIM_EFILE_ADN*** the Abbrv. Dialing Num file

TAPI_SIM_EFILE_FDN the Fixed Dialing Num file

TAPI_SIM_EFILE_SMS the Short Messages file

TAPI_SIM_EFILE_CCP the Capab. Config Param file

TAPI_SIM_EFILE_IMSI the IMSI file

TAPI_SIM_EFILE_SST the SIM Service Table file

TAPI_SIM_EFILE_EST the Extended Service Table File

TAPI_SIM_EFILE_ACM the Accumul. Call Meter file

TAPI_SIM_EFILE_MBDN SIM Mail Box Dailing Number file

TAPI_SIM_EFILE_GID1 the Group Identifier Level 1

TAPI_SIM_EFILE_GID2 the Group Identifier Level 2

TAPI_SIM_EFILE_MSISDN the MSISDN file

TAPI_SIM_EFILE_SMSP the Short Msg Srv. Param file

TAPI_SIM_EFILE_SMSS the SMS Status

TAPI_SIM_EFILE_EXT1 the Extension 1 file

TAPI_SIM_EFILE_EXT2 the Extension 2 file

TAPI_SIM_EFILE_LP the Language Preference file

TAPI_SIM_EFILE_ACM_MAX the ACM maximum value file

TAPI_SIM_EFILE_PUCT the Price per Unit and. file

TAPI_SIM_EFILE_CBMI the Cell Brd msg Id sel file

TAPI_SIM_EFILE_PHASE the PHASE identification file

TAPI_SIM_EFILE_SDN the Service Dialing Numbers

TAPI_SIM_EFILE_EXT3 the Extension 3 file

TAPI_SIM_EFILE_BDN the BDN file

TAPI_SIM_EFILE_EXT4 the Extension 4 file

TAPI_SIM_EFILE_ECC the Emergency Call Codes

TAPI_SIM_EFILE_SPN the Service Provider Name

TAPI_SIM_EFILE_PNN the Service Provider Name

TAPI_SIM_EFILE_IMG image

TAPI_SIM_EFILE_USIM_LI Language Indication

TAPI_SIM_EFILE_USIM_IMSI IMSI

TAPI_SIM_EFILE_USIM_ACM_MAX ACM maximum value

TAPI_SIM_EFILE_USIM_UST USIM Service Table

TAPI_SIM_EFILE_USIM_ACM Accumulated Call Meter

TAPI_SIM_EFILE_USIM_SPN Service Provider Name

TAPI_SIM_EFILE_USIM_PUCT Price per Unit and Currency Table

TAPI_SIM_EFILE_USIM_CBMI Cell Broadcaster Message Identifier selection

TAPI_SIM_EFILE_USIM_ECC Emergency Call Codes

TAPI_SIM_EFILE_USIM_FDN Fixed Dialing Number

TAPI_SIM_EFILE_USIM_SMS USIM SMS

TAPI_SIM_EFILE_USIM_MSISDN USIM MSISDN

TAPI_SIM_EFILE_USIM_SMSP USIM SMSP

TAPI_SIM_EFILE_USIM_SMSS USIM SMSS

TAPI_SIM_EFILE_USIM_SDN USIM SDN

TAPI_SIM_EFILE_USIM_EXT1 USIM EXT1

TAPI_SIM_EFILE_USIM_EXT2 USIM EXT2

TAPI_SIM_EFILE_USIM_EXT3 USIM EXT3

TAPI_SIM_EFILE_USIM_EXT4 USIM EXT4

TAPI_SIM_EFILE_USIM_EXT5 Extension 5

TAPI_SIM_EFILE_USIM_CCP2 Capability Configuration Parameters 2

TAPI_SIM_EFILE_USIM_HIDDEN_KEY Key for hidden phonebook entries

TAPI_SIM_EFILE_USIM_BDN Barred Dialing Numbers

TAPI_SIM_EFILE_USIM_MBDN USIM Mail box dialing number

TAPI_SIM_EFILE_USIM_MBI Mailbox Identifier

TAPI_SIM_EFILE_USIM_PHONEBOOK_PBR Phonebook Reference File

TAPI_SIM_EFILE_USIM_PHONEBOOK_CCP1 Capability Configuration Parameters 1

TAPI_SIM_EFILE_USIM_PHONEBOOK_UID unique identifier

TAPI_SIM_EFILE_USIM_PHONEBOOK_PSC phonebook synchronization counter

TAPI_SIM_EFILE_USIM_PHONEBOOK_CC change counter

TAPI_SIM_EFILE_USIM_PHONEBOOK_PUID previous unique identifier

TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN Abbreviated Dialing Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP Index Administration Phonebook

TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC Phonebook Control

TAPI_SIM_EFILE_USIM_PHONEBOOK_EXT1 Extension 1

TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP Grouping File

TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS Additional number Alpha String

TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS1 Additional number Alpha String

TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS1 Second Name Entry

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE Second Name Entry

TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL e-mail address

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR1 Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA1 Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB1 Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC1 Additional Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN1 Abbreviated Dialing Number

TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC1 Phonebook Control

TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP1 Grouping File

TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE1 Second Name Entry

TAPI_SIM_EFILE_USIM_PHONEBOOK_UID1 unique identifier

TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL1 e-mail address

TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP1 Index Administration Phonebook

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR Phonebook Reference File

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR1 Phonebook Reference File 1

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR2 Phonebook Reference File 1

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CCP1 Capability Configuration Parameters 1

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID unique identifier

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PSC Phonebook synchronization counter

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CC change counter

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PUID previous unique identifier

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN Abbreviated Dialing Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP Index Administration Phonebook

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC Phonebook Control

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EXT1 Extension 1

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP Grouping File

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_AAS Additional number Alpha String

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GAS Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE Second Name Entry

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL e-mail address

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRC Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR1 Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA1 Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB1 Additional Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN1 Abbreviated Dilling Number

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC1 Phonebook Control

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP1 Grouping File

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE1 Second Name Entry

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID1 unique identifier

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL1 e-mail address

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP1 Index Administration Phonebook

TAPI_SIM_EFILE_INVALID Elements enumerating (known) Elementary files in SIM like EF-DIR, EF-VGCS, EF-VBS, EF-PBR, EF-ADN, EF-IAP, EF-EXT1, EF-SNE, EF-ANR, EF-PBC, EF-GRP, EF-AAS, EF-GAS, EF-UID, EF-EMAIL, EF-CCP1, EF-PSC, EF-CC, EF-PUID, EF-FDN etc.

TAPI_SIM_EFILE_OTHERS Element to indicate an unknown file.

5.94.2.8 enum TelSimFileType_t

This enumeration defines the file type.

Enumerator:

TAPI_SIM_FTYPE_DEDICATED	Dedicated
TAPI_SIM_FTYPE_TRANSPARENT	Transparent
TAPI_SIM_FTYPE_LINEAR_FIXED	Linear fixed
TAPI_SIM_FTYPE_CYCLIC	Cyclic
TAPI_SIM_FTYPE_INVALID_TYPE	Invalid type

5.94.2.9 enum TelSimLangIndicationCode_t

This enumeration defines the language indication code.

Enumerator:

TAPI_SIM_LI_GERMAN	German
TAPI_SIM_LI_ENGLISH	English
TAPI_SIM_LI_ITALIAN	Italian
TAPI_SIM_LI_FRENCH	French
TAPI_SIM_LI_SPANISH	Spanish
TAPI_SIM_LI_DUTCH	Dutch
TAPI_SIM_LI_SWEDISH	Swedish
TAPI_SIM_LI_DANISH	Danish
TAPI_SIM_LI_PORTUGUESE	Portuguese
TAPI_SIM_LI_FINNISH	Finnish
TAPI_SIM_LI_NORWEGIAN	Norwegian
TAPI_SIM_LI_GREEK	Greek
TAPI_SIM_LI_TURKISH	Turkish
TAPI_SIM_LI_HUNGARIAN	Hungarian
TAPI_SIM_LI_POLISH	Posish
TAPI_SIM_LI_LANG_UNSPECIFIED	Unspecified

5.94.2.10 enum TelSimNumberingPlanIdentity_t

This enumeration defines the numbering plan identifier.

Enumerator:

TAPI_SIM_NPI_UNKNOWN	Unknown
TAPI_SIM_NPI_ISDN_TEL	ISDN/Telephone numbering plan
TAPI_SIM_NPI_DATA_NUMBERING_PLAN	Data numbering plan
TAPI_SIM_NPI_TELEX	Telex numbering plan
TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN	Service Center Specific plan

TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN2 Service Center Specific plan

TAPI_SIM_NPI_NATIONAL National numbering plan

TAPI_SIM_NPI_PRIVATE Private numbering plan

TAPI_SIM_NPI_ERMES_NUMBERING_PLAN ERMES numbering plan

TAPI_SIM_NPI_RESERVED_FOR_EXT Reserved for extension

5.94.2.11 enum TelSimPbAccessMode_t

This enumeration defines modes to add,del,edit for phone book access

Enumerator:

TAPI_SIM_PB_ACCESS_MODE_ADD Phone Book Access mode - Add entry

TAPI_SIM_PB_ACCESS_MODE_DEL Phone Book Access mode - Delete entry

TAPI_SIM_PB_ACCESS_MODE_EDIT Phone Book Access mode - edit entry

5.94.2.12 enum TelSimPbAccessResult_t

This enumeration defines the Phone book access result

Enumerator:

TAPI_SIM_PB_SUCCESS SIM phonebook operation successful.

TAPI_SIM_PB_FAIL SIM phonebook operation failure.

TAPI_SIM_PB_INVALID_INDEX The index passed was not a valid index.

TAPI_SIM_PB_INVALID_NUMBER_LENGTH The number length is exceecds the max length allowed (or 0).

TAPI_SIM_PB_INVALID_NAME_LENGTH The name length is exceecds the max length allowed (or 0).

TAPI_SIM_PB_ACCESS_CONDITION_NOT_SATISFIED Access condition for PB file is not satisfied.

5.94.2.13 enum TelSimPbFileType_t

This enumeration defines different storage types to be selected in SIM or USIM

Enumerator:

TAPI_SIM_PB_FDN sim FDN

TAPI_SIM_PB_LDN sim last dialled num

TAPI_SIM_PB_MSISDN sim my own number - MSISDN list

TAPI_SIM_PB_ADN SIM - ADN

TAPI_SIM_PB_SDN Sim service dialing num

5.94.2.14 enum TelSimPinOperationResult_t

This enumeration defines the pin operation result from the lower layers.

Enumerator:

TAPI_SIM_PIN_OPERATION_SUCCESS Operation involving PIN (verification/change/enable/disable, etc) is successful.

TAPI_SIM_BUSY_STATUS Sim is busy

TAPI_SIM_CARD_ERROR_STATUS Sim card error

TAPI_SIM_INCOMPATIBLE_PIN_OPERATION_STATUS Sim Incompatible pin operation that is in case when invalid SIM command is given or incorrect parameters are supplied to the SIM.

TAPI_SIM_PIN_INCORRECT_PASSWORD_STATUS Sim PIN Incorrect password

TAPI_SIM_PIN2_REQD_STATUS Sim PIN2 Required Status

TAPI_SIM_PIN2_INCORRECT_PASSWORD_STATUS Sim PIN2 Incorrect password

TAPI_SIM_PUK_INCORRECT_PASSWORD_STATUS Sim PUK Incorrect Password

TAPI_SIM_PUK_REQD_STATUS PUK Required

TAPI_SIM_PUK2_REQD_STATUS PUK2 Required

5.94.2.15 enum TelSimPinStatus_t

This enumeration defines the pin status.

Enumerator:

TAPI_SIM_PIN_STATUS_DISABLED PIN status Disabled

TAPI_SIM_PIN_STATUS_ENABLED PIN status Enabled

TAPI_SIM_PIN_STATUS_BLOCKED SIM is present, but PIN is blocked. Need unblocking by PUK

TAPI_SIM_PIN_STATUS_PUK_BLOCKED SIM is present, but PUK is blocked

TAPI_SIM_PIN_STATUS_UNKNOWN SIM is in unknown state

5.94.2.16 enum TelSimPinType_t

This enumeration defines the pin type.

Enumerator:

TAPI_SIM_PTYPE_PIN1 PIN 1

TAPI_SIM_PTYPE_PIN2 PIN 2

TAPI_SIM_PTYPE_PUK1 Puk 1

TAPI_SIM_PTYPE_PUK2 Puk 2

TAPI_SIM_PTYPE_UPIN Universal PIN

TAPI_SIM_PTYPE_ADM Administrator

5.94.2.17 enum TelSimRecordMode_t

This enumeration defines the record mode.

Enumerator:

TAPI_SIM_RECORD_NONE N/A, meaning the target file isn't a record type

TAPI_SIM_RECORD_ABSOLUTE The specified record will be read or updated

TAPI_SIM_RECORD_CURRENT The current record will be read or updated

TAPI_SIM_RECORD_NEXT The next record will be read or updated

TAPI_SIM_RECORD_PREVIOUS The previous record will be read or updated

TAPI_SIM_RECORD_TO_THE_LAST All records from the record specified by record index to the last one will be read to the last

TAPI_SIM_RECORD_FROM_THE_FIRST All records from the first record to the one specified by record_index will be read from the first

5.94.2.18 enum TelSimTextEncrypt_t

This enumertion defines the text encryption types

Enumerator:

TAPI_SIM_TEXT_ENC_ASCII ASCII Encoding

TAPI_SIM_TEXT_ENC_GSM7BIT GSM 7 Bit Encoding

TAPI_SIM_TEXT_ENC_UCS2 UCS2 Encoding

TAPI_SIM_TEXT_ENC_HEX HEX Encoding

5.94.2.19 enum TelSimTypeOfNum_t

This enumeration defines the type of number.

Enumerator:

TAPI_SIM_TON_UNKNOWN unknown

TAPI_SIM_TON_INTERNATIONAL international number

TAPI_SIM_TON_NATIONAL national number

TAPI_SIM_TON_NETWORK_SPECIFIC network specific number

TAPI_SIM_TON_DEDICATED_ACCESS subscriber number

TAPI_SIM_TON_ALPHA_NUMERIC alphanumeric, GSM 7-bit default alphabet)

TAPI_SIM_TON_ABBREVIATED_NUMBER abbreviated number

TAPI_SIM_TON_RESERVED_FOR_EXT reserved for extension

5.95 SS DataStructures

Files

- file TelSs.h

  This file serves as a "C" header file defines structures for Tapi SS Services.
  It contains a sample set of constants, enums,structs that would be required by applications.

Data Structures

- struct TelSsForwardInfo_t
- struct TelSsUssdMsgInfo_t
- struct TelSsCallBarringInfo_t
- struct TelSsCliRecord_t
- struct TelSsCallBarringRecord_t
- struct TelSsForwardRecord_t
- struct TelSsCallWaitingRecord_t
- union TelSsRecord_t
- struct TelSsInfo_t
- struct TelSsCapabilityInfo_t

Defines

- #define TAPI_SS_GSM_BARR_PW_LEN_MAX 4
- #define TAPI_CALL_DIALDIGIT_LEN_MAX 82
- #define TAPI_SS_USSD_DATA_SIZE_MAX 201

Enumerations

- enum TelSsForwardWhen_t {

  TAPI_SS_FORWARD_WHEN_UNCONDITIONAL_EV, TAPI_SS_FORWARD_WHEN_BUSY_EV, TAPI_SS_FORWARD_WHEN_NO_ANSWER_EV, TAPI_SS_FORWARD_WHEN_NOT_REACHABLE_EV,

  TAPI_SS_FORWARD_WHEN_ALL_FORWARDING_EV }
- enum TelSsCallBarType_t {

  TAPI_CALL_BARRING_ALL, TAPI_CALL_BARRING_OUTGOING, TAPI_CALL_BARRING_ALL_OUTGOING_CALLS, TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL,

  TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL_EXCEPT, TAPI_CALL_BARRING_INCOMING, TAPI_CALL_BARRING_ALL_INCOMING_CALLS, TAPI_CALL_BARRING_ALL_INCOMING_CALLS_ROAMING,

  TAPI_CALL_BARRING_ALL_INCOMING_CALLS_INSIM }
- enum TelSsTeleService_t {

  TAPI_SS_TS_ALL_TELE_AND_BEARER_SERVICES, TAPI_SS_TS_ALL_TELESERVICES, TAPI_SS_TS_ALL_SPEECH, TAPI_SS_TS_ALL_DATA_TELESERVICES,

  TAPI_SS_TS_FACS, TAPI_SS_TS_ALL_TELESERVICES_EXCEPT_SMS, TAPI_SS_TS_ALL_PLMN_SPEC_TELE, TAPI_SS_TS_PLMN_SPEC_TELE_1, TAPI_SS_TS_PLMN_SPEC_TELE_2, TAPI_SS_TS_PLMN_SPEC_TELE_3, TAPI_SS_TS_PLMN_SPEC_TELE_4, TAPI_SS_TS_PLMN_SPEC_TELE_5,

TAPI_SS_TS_PLMN_SPEC_TELE_6, TAPI_SS_TS_PLMN_SPEC_TELE_7, TAPI_SS_TS_PLMN_SPEC_TELE_8, TAPI_SS_TS_PLMN_SPEC_TELE_9,

TAPI_SS_TS_PLMN_SPEC_TELE_A, TAPI_SS_TS_PLMN_SPEC_TELE_B, TAPI_SS_TS_PLMN_SPEC_TELE_C, TAPI_SS_TS_PLMN_SPEC_TELE_D,

TAPI_SS_TS_PLMN_SPEC_TELE_E, TAPI_SS_TS_PLMN_SPEC_TELE_F, **TAPI_SS_TS_PLMN_SPEC_TELE_**, TAPI_SS_BS_ALL_BEARER_SERVICES,

TAPI_SS_BS_ALL_DATA_CDS, TAPI_SS_BS_ALL_ASYNCHRONOUS, TAPI_SS_BS_ALL_SYNCHRONOUS, TAPI_SS_BS_DATA_CIRCUIT_SYNC,

TAPI_SS_BS_DATA_CIRCUIT_ASYNC, TAPI_SS_BS_DATA_ALL_PDS, TAPI_SS_BS_ALL_DATA_PADCA, TAPI_SS_BS_ALL_PLMN_SPEC_BS,

TAPI_SS_BS_PLMN_SPEC_TELE_1, TAPI_SS_BS_PLMN_SPEC_TELE_2, TAPI_SS_BS_PLMN_SPEC_TELE_3, TAPI_SS_BS_PLMN_SPEC_TELE_4,

TAPI_SS_BS_PLMN_SPEC_TELE_5, TAPI_SS_BS_PLMN_SPEC_TELE_6, TAPI_SS_BS_PLMN_SPEC_TELE_7, TAPI_SS_BS_PLMN_SPEC_TELE_8,

TAPI_SS_BS_PLMN_SPEC_TELE_9, TAPI_SS_BS_PLMN_SPEC_TELE_A, TAPI_SS_BS_PLMN_SPEC_TELE_B, TAPI_SS_BS_PLMN_SPEC_TELE_C,

TAPI_SS_BS_PLMN_SPEC_TELE_D, TAPI_SS_BS_PLMN_SPEC_TELE_E, TAPI_SS_BS_PLMN_SPEC_TELE_F, TAPI_SS_TS_ALL_DATA,

TAPI_SS_TS_DATA_CIRCUIT_SYNC, TAPI_SS_TS_MULTIPLE, TAPI_SS_TS_TELEPHONY, TAPI_SS_TS_DATA_CIRCUIT_ASYNC,

TAPI_SS_TS_DATA_PACKET }

- enum TelSsUssdRespType_t { TAPI_USSD_RES_UNKNOWN_ALPHABET, TAPI_USSD_RES_DISPL_BUSY, TAPI_USSD_RES_OK, TAPI_USSD_RES_NONE }
- enum TelSsUssdIndType_t {

  TAPI_USSD_IND_NOTIFY = 0x00, TAPI_USSD_IND_NO_ACTION_REQUIRE = 0x01, TAPI_USSD_IND_ACTION_REQUIRE = 0x02, TAPI_USSD_IND_TERMINATED_BY_NET = 0x03,

  TAPI_USSD_IND_OTHER_CLIENT = 0x04, TAPI_USSD_IND_NOT_SUPPORT = 0x05, TAPI_USSD_IND_TIME_OUT = 0x06 }
- enum TelSsCallBarOperationMode_t { TAPI_SS_CALL_BAR_ACTIVATE, TAPI_SS_CALL_BAR_DEACTIVATE }
- enum TelSsCallType_t {

  TAPI_CALL_TYPE_VOICE_EV, TAPI_CALL_TYPE_DATA_EV, TAPI_CALL_TYPE_FAX_EV, TAPI_CALL_TYPE_SHORT_MESS_EV,

  TAPI_CALL_TYPE_DATA_CIRCUIT_SYNC_EV }
- enum TelSsForwardMode_t { TAPI_CALL_FORWARD_MODE_DISABLE_EV, TAPI_CALL_FORWARD_MODE_ENABLE_EV, TAPI_CALL_FORWARD_MODE_REGISTRATION_EV, TAPI_CALL_FORWARD_MODE_ERASURE_EV }
- enum TelSsForwardType_t { TAPI_CS_FORWARD_TYPE_VOICE_EV = 0, TAPI_CS_FORWARD_TYPE_DATA_EV, TAPI_CS_FORWARD_TYPE_FAX_EV }
- enum TelSsRestrictIDStatus_t { TAPI_CALL_RESTRICT_ID_DEPEND_NETWORK_EV, TAPI_CALL_RESTRICT_ID_ENABLE_EV, TAPI_CALL_RESTRICT_ID_DISABLE_EV }
- enum TelSsType_t {

  TAPI_SS_TYPE_BARRING, TAPI_SS_TYPE_FORWARDING, TAPI_SS_TYPE_WAITING, TAPI_SS_TYPE_IDENTIFICATION,

  TAPI_SS_TYPE_CNAP, TYPE_SS_TYPE_USSD }

- enum TelSsCause_t {

  TAPI_SS_SUCCESS, TAPI_SS_TIMEREXPIRE, TAPI_SS_UNKNOWNSUBSCRIBER, TAPI_SS_SERVICENOTPROVISIONED,

  TAPI_SS_CALLBARRED, TAPI_SS_ILLEGALSSOPERATION_ERROR, TAPI_SS_NOTAVAILABLE, TAPI_SS_SUBSCRIPTIONVIOLATION,

  TAPI_SS_FACILITYNOTSUPPORTED, TAPI_SS_UNEXPECTEDDATAVALUE, TAPI_SS_PWREGISTRATIONFAILURE, TAPI_SS_NEGATIVEPWCHECK,

  TAPI_SS_NUMBEROFPWATTEMPTSVIOLATION, TAPI_SS_UNKNOWNALPHABET, TAPI_SS_USSDBUSY, TAPI_SS_REJECTEDBYUSER,

  TAPI_SS_REJECTEDBYNETWORK, TAPI_SS_INVALIDDEFLECTEDTONUMBER, TAPI_SS_MAXNOMPTYEXCEEDED, TAPI_SS_RESOURCESNOTAVAILABLE,

  TAPI_SS_UNKNOWNERROR, TAPI_SS_OEM_NOT_SUPPORTED }
- enum TelSsStatus_t {

  TAPI_SS_STATUS_ACTIVE, TAPI_SS_STATUS_REGISTERED, TAPI_SS_STATUS_PROVISIONED, TAPI_SS_STATUS_QUIESCENT,

  TAPI_SS_STATUS_NOTHING }
- enum TelSsLineIdentificationType_t {

  TAPI_CALLING_LINE_IDENTITY_PRESENTATION, TAPI_CALLING_LINE_IDENTITY_RESTRICTION, TAPI_CONNECTED_LINE_IDENTITY_PRESENTATION, TAPI_CONNECTED_LINE_IDENTITY_RESTRICTION,

  TAPI_CALLED_LINE_IDENTITY_PRESENTATION, TAPI_CALLING_NAME_PRESENTATION }
- enum TelSsCliStatus_t {

  TAPI_CLI_STATUS_NOT_PROVISONED, TAPI_CLI_STATUS_PROVISIONED, TAPI_CLI_STATUS_ACTIVATED, TAPI_CLI_STATUS_UNKOWN,

  TAPI_CLI_STATUS_TEMP_RESTRICTED, TAPI_CLI_STATUS_TEMP_ALLOWED }
- enum TelSsBarPwdType_t { TAPI_SS_BAR_GET_PWD, TAPI_SS_BAR_NEW_PWD, TAPI_SS_BAR_VERIFY_NEW_PWD }

5.95.1 Define Documentation

5.95.1.1 #define TAPI_CALL_DIALDIGIT_LEN_MAX 82

maximum length of dial digit

5.95.1.2 #define TAPI_SS_GSM_BARR_PW_LEN_MAX 4

maximum length of barring password

5.95.1.3 #define TAPI_SS_USSD_DATA_SIZE_MAX 201

User-to-user data maximum size

5.95.2 Enumeration Type Documentation

5.95.2.1 enum TelSsBarPwdType_t

This enumeration defines the Bar password type.

Enumerator:

TAPI_SS_BAR_GET_PWD This password type indicates already set barring password

TAPI_SS_BAR_NEW_PWD This password type indicates new password

TAPI_SS_BAR_VERIFY_NEW_PWD This password type indicates verifying new password

5.95.2.2 enum TelSsCallBarOperationMode_t

This enumeration defines the call barring operation mode .

Enumerator:

TAPI_SS_CALL_BAR_ACTIVATE Activate call barring

TAPI_SS_CALL_BAR_DEACTIVATE De Activate call barring

5.95.2.3 enum TelSsCallBarType_t

The various types of call barring. Note, only one type of incoming barring and only one type of outgoing barring may be active at any time.

Enumerator:

TAPI_CALL_BARRING_ALL Barring all outgoing and incoming calls

TAPI_CALL_BARRING_OUTGOING Barring outgoing call

TAPI_CALL_BARRING_ALL_OUTGOING_CALLS Barring all outgoing calls

TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL Barring all outgoing international calls

TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL_EXCEPT Barring all international outgoing

TAPI_CALL_BARRING_INCOMING Barring the incoming call

TAPI_CALL_BARRING_ALL_INCOMING_CALLS Barring all incoming calls

TAPI_CALL_BARRING_ALL_INCOMING_CALLS_ROAMING Barring all incoming calls while roaming

TAPI_CALL_BARRING_ALL_INCOMING_CALLS_INSIM Barring Incoming Calls from numbers which is not stored in the SIM memory

5.95.2.4 enum TelSsCallType_t

This enumeration defines the call type .

Enumerator:

TAPI_CALL_TYPE_VOICE_EV Voice Call

TAPI_CALL_TYPE_DATA_EV Data Call

TAPI_CALL_TYPE_FAX_EV Fax Call

TAPI_CALL_TYPE_SHORT_MESS_EV Short Message

TAPI_CALL_TYPE_DATA_CIRCUIT_SYNC_EV Data Circuit Sync

5.95.2.5 enum TelSsCause_t

Describes whether a supplementary service request was a success or a failure; and if it failed, why.

Enumerator:

TAPI_SS_SUCCESS SS operation was successful

TAPI_SS_TIMEREXPIRE SS operation timer expired on network.

TAPI_SS_UNKNOWNSUBSCRIBER SS error indicating unknown/illegal subscriber.

TAPI_SS_SERVICENOTPROVISIONED SS service not provisioned

TAPI_SS_CALLBARRED SS operation barred.

TAPI_SS_ILLEGALSSOPERATION_ERROR Illegal SS operation

TAPI_SS_NOTAVAILABLE SS not available in network

TAPI_SS_SUBSCRIPTIONVIOLATION SS service subscription violation.

TAPI_SS_FACILITYNOTSUPPORTED SS service facility not supported

TAPI_SS_UNEXPECTEDDATAVALUE SS error indicating unexpected data value on network side

TAPI_SS_PWREGISTRATIONFAILURE SS error indicating change password failure.

TAPI_SS_NEGATIVEPWCHECK SS error indicating negative password check.

TAPI_SS_NUMBEROFPWATTEMPTSVIOLATION SS error indicating barring password attempts violated.

TAPI_SS_UNKNOWNALPHABET SS error indicating unknown SS data coding of alphabet

TAPI_SS_USSDBUSY SS error indicating USSD Busy(Already ss / ussd is ongoing).

TAPI_SS_REJECTEDBYUSER SS operation rejected by user.

TAPI_SS_REJECTEDBYNETWORK SS operation rejected by network.

TAPI_SS_INVALIDDEFLECTEDTONUMBER SS error indicating the invalid deflected to number.

TAPI_SS_MAXNOMPTYEXCEEDED SS error indicating Maximum mpty is reached.

TAPI_SS_RESOURCESNOTAVAILABLE SS error indicating resources not available in network.

TAPI_SS_UNKNOWNERROR SS error indicating unknown error

TAPI_SS_OEM_NOT_SUPPORTED If oem do not support any of SS requests, then this error will be returned back

5.95.2.6 enum TelSsCliStatus_t

This enumeration defines the cli service status .

Enumerator:

TAPI_CLI_STATUS_NOT_PROVISONED Service not provided by the service provider

TAPI_CLI_STATUS_PROVISIONED Service is provided by the service provider

TAPI_CLI_STATUS_ACTIVATED Service is activated at the network

TAPI_CLI_STATUS_UNKOWN Service status is unknown

TAPI_CLI_STATUS_TEMP_RESTRICTED Service is temporarily restricted

TAPI_CLI_STATUS_TEMP_ALLOWED Service is temporarily allowed

5.95.2.7 enum TelSsForwardMode_t

This enumeration defines the forward mode .

Enumerator:

TAPI_CALL_FORWARD_MODE_DISABLE_EV Deactivate call forwarding

TAPI_CALL_FORWARD_MODE_ENABLE_EV Activate call forwarding

TAPI_CALL_FORWARD_MODE_REGISTRATION_EV Register Call forwarding

TAPI_CALL_FORWARD_MODE_ERASURE_EV De-register call forwarding

5.95.2.8 enum TelSsForwardType_t

This enumeration defines the forward type .

Enumerator:

TAPI_CS_FORWARD_TYPE_VOICE_EV Voice call forward

TAPI_CS_FORWARD_TYPE_DATA_EV Data call forward

TAPI_CS_FORWARD_TYPE_FAX_EV Fax call forward

5.95.2.9 enum TelSsForwardWhen_t

/∗∗ This enumeration defines the call forwarding flavor.

Enumerator:

TAPI_SS_FORWARD_WHEN_UNCONDITIONAL_EV Unconditional

TAPI_SS_FORWARD_WHEN_BUSY_EV When Busy

TAPI_SS_FORWARD_WHEN_NO_ANSWER_EV When No Reply

TAPI_SS_FORWARD_WHEN_NOT_REACHABLE_EV When Not Reachable

TAPI_SS_FORWARD_WHEN_ALL_FORWARDING_EV Forward All Calls

5.95.2.10 enum TelSsLineIdentificationType_t

These are the four types of identity presentation / restriction services.

Enumerator:

TAPI_CALLING_LINE_IDENTITY_PRESENTATION identify the party calling this phone

TAPI_CALLING_LINE_IDENTITY_RESTRICTION hide the identity of this phone when calling others

TAPI_CONNECTED_LINE_IDENTITY_PRESENTATION identify the party to whom the calling party (this phone) is connected

TAPI_CONNECTED_LINE_IDENTITY_RESTRICTION restrict yourself from being identified by incoming calls, such as forwarded calls

TAPI_CALLED_LINE_IDENTITY_PRESENTATION Called line identity presentation

TAPI_CALLING_NAME_PRESENTATION Calling Name Presentation

5.95.2.11 enum TelSsRestrictIDStatus_t

This enumeration defines the types of restrict identification .

Enumerator:

TAPI_CALL_RESTRICT_ID_DEPEND_NETWORK_EV Depend on Network

TAPI_CALL_RESTRICT_ID_ENABLE_EV Enable Restricted Call Id

TAPI_CALL_RESTRICT_ID_DISABLE_EV Disable Restricted Call Id

5.95.2.12 enum TelSsStatus_t

The status of a supplementary service feature (e.g. Call Forwarding or Call Barring).

These enumerated values should be used as masks

Enumerator:

TAPI_SS_STATUS_ACTIVE Active Status

TAPI_SS_STATUS_REGISTERED SS Registered

TAPI_SS_STATUS_PROVISIONED Provision established

TAPI_SS_STATUS_QUIESCENT Provision established

TAPI_SS_STATUS_NOTHING Status Nothing

5.95.2.13 enum TelSsTeleService_t

The types of telecommunication services are used for setting the statuses of supplementary services. They are typically not supported in IS-2000 or IS-136. See GSM 02.30 Annex C and GSM 22.030 Annex C.

Enumerator:

TAPI_SS_TS_ALL_TELE_AND_BEARER_SERVICES All Teleservices & Bearer Services

TAPI_SS_TS_ALL_TELESERVICES All Teleservices

TAPI_SS_TS_ALL_SPEECH All Speech Services

TAPI_SS_TS_ALL_DATA_TELESERVICES All Data Teleservices

TAPI_SS_TS_FACS All Fax Service

TAPI_SS_TS_ALL_TELESERVICES_EXCEPT_SMS All teleservice except SMS

TAPI_SS_TS_ALL_PLMN_SPEC_TELE PLMN specific teleservices

TAPI_SS_TS_PLMN_SPEC_TELE_1 PLMN specific teleservice 1

TAPI_SS_TS_PLMN_SPEC_TELE_2 PLMN specific teleservice 2

TAPI_SS_TS_PLMN_SPEC_TELE_3 PLMN specific teleservice 3

TAPI_SS_TS_PLMN_SPEC_TELE_4 PLMN specific teleservice 4

TAPI_SS_TS_PLMN_SPEC_TELE_5 PLMN specific teleservice 5

TAPI_SS_TS_PLMN_SPEC_TELE_6 PLMN specific teleservice 6

TAPI_SS_TS_PLMN_SPEC_TELE_7 PLMN specific teleservice 7

TAPI_SS_TS_PLMN_SPEC_TELE_8 PLMN specific teleservice 8

TAPI_SS_TS_PLMN_SPEC_TELE_9 PLMN specific teleservice 9

TAPI_SS_TS_PLMN_SPEC_TELE_A PLMN specific teleservice 10

TAPI_SS_TS_PLMN_SPEC_TELE_B PLMN specific teleservice 11

TAPI_SS_TS_PLMN_SPEC_TELE_C PLMN specific teleservice 12

TAPI_SS_TS_PLMN_SPEC_TELE_D PLMN specific teleservice 13

TAPI_SS_TS_PLMN_SPEC_TELE_E PLMN specific teleservice 14

TAPI_SS_TS_PLMN_SPEC_TELE_F PLMN specific teleservice 15

TAPI_SS_BS_ALL_BEARER_SERVICES All PLMN specific bearer services

TAPI_SS_BS_ALL_DATA_CDS All Data CDS

TAPI_SS_BS_ALL_ASYNCHRONOUS All Asynchronous

TAPI_SS_BS_ALL_SYNCHRONOUS All Synchronous

TAPI_SS_BS_DATA_CIRCUIT_SYNC All Data Circuit Synchronous

TAPI_SS_BS_DATA_CIRCUIT_ASYNC All Data Circuit Asynchronous

TAPI_SS_BS_DATA_ALL_PDS All Data PDS

TAPI_SS_BS_ALL_DATA_PADCA All Data PADCA

TAPI_SS_BS_ALL_PLMN_SPEC_BS All PLMN Spec BS

TAPI_SS_BS_PLMN_SPEC_TELE_1 PLMN specific bearer service 1

TAPI_SS_BS_PLMN_SPEC_TELE_2 PLMN specific bearer service 2

TAPI_SS_BS_PLMN_SPEC_TELE_3 PLMN specific bearer service 3

TAPI_SS_BS_PLMN_SPEC_TELE_4 PLMN specific bearer service 4

TAPI_SS_BS_PLMN_SPEC_TELE_5 PLMN specific bearer service 5

TAPI_SS_BS_PLMN_SPEC_TELE_6 PLMN specific bearer service 6

TAPI_SS_BS_PLMN_SPEC_TELE_7 PLMN specific bearer service 7

TAPI_SS_BS_PLMN_SPEC_TELE_8 PLMN specific bearer service 8

TAPI_SS_BS_PLMN_SPEC_TELE_9 PLMN specific bearer service 9

TAPI_SS_BS_PLMN_SPEC_TELE_A PLMN specific bearer service 10

TAPI_SS_BS_PLMN_SPEC_TELE_B PLMN specific bearer service 11

TAPI_SS_BS_PLMN_SPEC_TELE_C PLMN specific bearer service 12

TAPI_SS_BS_PLMN_SPEC_TELE_D PLMN specific bearer service 13

TAPI_SS_BS_PLMN_SPEC_TELE_E PLMN specific bearer service 14

TAPI_SS_BS_PLMN_SPEC_TELE_F PLMN specific bearer service 15

TAPI_SS_TS_ALL_DATA All Data

TAPI_SS_TS_DATA_CIRCUIT_SYNC Data Circuit Sync

TAPI_SS_TS_MULTIPLE multiple

TAPI_SS_TS_TELEPHONY Telephony

TAPI_SS_TS_DATA_CIRCUIT_ASYNC Data Circuit Async

TAPI_SS_TS_DATA_PACKET Data packet

5.95.2.14 enum TelSsType_t

This enumeration type defines the Supplementary service types

Enumerator:

TAPI_SS_TYPE_BARRING Barring SS

TAPI_SS_TYPE_FORWARDING Forward SS

TAPI_SS_TYPE_WAITING Waiting SS

TAPI_SS_TYPE_IDENTIFICATION Identification SS

TAPI_SS_TYPE_CNAP CNAP SS

TYPE_SS_TYPE_USSD Unstructured supplementary data SS

5.95.2.15 enum TelSsUssdIndType_t

This enumeration defines the ussd Indication Type

Enumerator:

TAPI_USSD_IND_NOTIFY notify : to display ussd data to user

TAPI_USSD_IND_NO_ACTION_REQUIRE No further user action required

TAPI_USSD_IND_ACTION_REQUIRE Further user action required

TAPI_USSD_IND_TERMINATED_BY_NET 0x03 : USSD terminated by network

TAPI_USSD_IND_OTHER_CLIENT 0x04 : other local client has responded

TAPI_USSD_IND_NOT_SUPPORT 0x05 : operation not supported

TAPI_USSD_IND_TIME_OUT 0x06 : Time out when there is no response from network

5.95.2.16 enum TelSsUssdRespType_t

This enumeration defines the ussd response type

Enumerator:

TAPI_USSD_RES_UNKNOWN_ALPHABET USSD data is unknown

TAPI_USSD_RES_DISPL_BUSY USSD Data response is display busy

TAPI_USSD_RES_OK USSD Data is encoded and properly propagated

TAPI_USSD_RES_NONE NO USSD Response from client

5.96 Utility Data Structures

Files

- file TelUtility.h

  This file serves as a "C" header file defines structures for Utility Services.
  It contains a sample set of constants, enums,structs that would be required by applications.

Data Structures

- struct TelTapiVersionInfo_t
- struct TelTapiTunnelInfo_t
- struct TelTapiErrorInfo_t

Enumerations

- enum TapiResult_t {

  TAPI_API_SUCCESS, TAPI_API_INVALID_INPUT, TAPI_API_INVALID_PTR, TAPI_API_INVALID_CALL_STATE,

  TAPI_API_NOT_SUPPORTED, TAPI_API_DEPRICATED, TAPI_API_SYSTEM_OUT_OF_MEM, TAPI_API_SYSTEM_RPC_LINK_DOWN,

  TAPI_API_MODEM_POWERED_OFF, TAPI_API_SERVICE_NOT_READY, TAPI_API_SERVER_FAILURE, TAPI_API_OEM_PLUGIN_FAILURE,

  TAPI_API_INVALID_CALL_HANDLE, TAPI_API_INVALID_DATA_LEN, TAPI_API_SIM_NOT_FOUND, TAPI_API_SIM_BUSY,

  TAPI_API_SIM_INVALID_CARD_TYPE, TAPI_API_SIM_CARD_PERMANENT_ERROR, TAPI_API_SIM_NOT_INITIALIZED, TAPI_API_REQUEST_MAX_IN_PROGRESS,

  TAPI_API_OFFLINE_MODE_ERROR, TAPI_EVENT_CLASS_UNKNOWN, TAPI_EVENT_UNKNOWN, TAPI_REGISTRATION_FAILURE }

5.96.1 Enumeration Type Documentation

5.96.1.1 enum TapiResult_t

Below enumerations are the return codes of TAPI API's

Enumerator:

TAPI_API_SUCCESS No Error occurred

TAPI_API_INVALID_INPUT Input values are not correct in TAPI Library

TAPI_API_INVALID_PTR Invalid pointer

TAPI_API_INVALID_CALL_STATE request failed due to invalid call state

TAPI_API_NOT_SUPPORTED The feature corresponding to requested API is not supported. This may be due to market/network/vendor reasons such as the feature is not available in the network.

TAPI_API_DEPRICATED This API is deprecated and will be so in future also

TAPI_API_SYSTEM_OUT_OF_MEM Out of memory

TAPI_API_SYSTEM_RPC_LINK_DOWN RPC link down

TAPI_API_MODEM_POWERED_OFF The Modem is powered off

TAPI_API_SERVICE_NOT_READY Phone was powered on, but yet to receive the power up completed notification

TAPI_API_SERVER_FAILURE error occurred in Telephony server

TAPI_API_OEM_PLUGIN_FAILURE Plug-in layer failure

TAPI_API_INVALID_CALL_HANDLE Invalid call handle

TAPI_API_INVALID_DATA_LEN Invalid data length

TAPI_API_SIM_NOT_FOUND Sim is not present / removed

TAPI_API_SIM_BUSY SIM is busy

TAPI_API_SIM_INVALID_CARD_TYPE Sim card is unknown

TAPI_API_SIM_CARD_PERMANENT_ERROR SIM error/blocked state

TAPI_API_SIM_NOT_INITIALIZED SIM has not initialized yet (waiting for PIN verification, etc)

TAPI_API_REQUEST_MAX_IN_PROGRESS Maximum number of API Request for the same service are already in progress

TAPI_API_OFFLINE_MODE_ERROR OEM Telephony Provider is in Offline mode.

TAPI_EVENT_CLASS_UNKNOWN Event class specified is not present in Event Class list.

TAPI_EVENT_UNKNOWN Event specified is not present in TAPI Unsolicited Event list.

TAPI_REGISTRATION_FAILURE Callback Registration/De-registration failed

Chapter 6

Data Structure Documentation

6.1 TelCallCugInfo_t Struct Reference

`#include <TelCall.h>`

6.1.1 Detailed Description

This data structure defines the CUG information.

Data Fields

- BOOL bCugFlag
- unsigned int Index
- TelCallCugOptions_t Option

6.1.2 Field Documentation

6.1.2.1 BOOL TelCallCugInfo_t::bCugFlag

Closed User Group flag - it holds TRUE when it has data and FALSE otherwise.

6.1.2.2 unsigned int TelCallCugInfo_t::Index

CUG Index value

6.1.2.3 TelCallCugOptions_t TelCallCugInfo_t::Option

Cug option type

The documentation for this struct was generated from the following file:

- TelCall.h

6.2 TelCallEndInfo_t Struct Reference

`#include <TelCall.h>`

6.2.1 Detailed Description

This structure contains information about Call end informationl.

Data Fields

- TelCallCause_t CallEndCause
- unsigned int pCallHandle
- unsigned int CallStartTime
- unsigned int CallEndTime

6.2.2 Field Documentation

6.2.2.1 TelCallCause_t TelCallEndInfo_t::CallEndCause

End cause for the call indicates whether the call is released normally or due to other cause

6.2.2.2 unsigned int TelCallEndInfo_t::pCallHandle

Call handle

6.2.2.3 unsigned int TelCallEndInfo_t::CallStartTime

Start time of the call(time in secs)

6.2.2.4 unsigned int TelCallEndInfo_t::CallEndTime

End time of the call(time in secs)

The documentation for this struct was generated from the following file:

- TelCall.h

6.3 TelCallIncomingCallInfo_t Struct Reference

```
#include <TelCall.h>
```

6.3.1 Detailed Description

This structure contains the incoming call information.

Data Fields

- unsigned int CallHandle
- TelCallType_t CallType
- TelSsLineIdentificationType_t CliPresentationIndicator
- char szCallingPartyNumber [TAPI_CALL_DIALDIGIT_LEN_MAX+1]
- char szCallingPartySubAddress [TAPI_CALL_SUBADDRESS_LEN_MAX+1]
- TelCallingNameInfo_t CallingNameInfo
- TelCallUusInfo_t UusInfo
- TelCallAlertingPattern_t AlertingPattern
- TelCallRedirectionInfo_t RedirectInfo
- int Signal

6.3.2 Field Documentation

6.3.2.1 unsigned int TelCallIncomingCallInfo_t::CallHandle

Handle for the call

6.3.2.2 TelCallType_t TelCallIncomingCallInfo_t::CallType

Call type

6.3.2.3 TelSsLineIdentificationType_t TelCallIncomingCallInfo_t::CliPresentationIndicator

Calling Party presentation indicator

6.3.2.4 char TelCallIncomingCallInfo_t::szCallingPartyNumber[TAPI_CALL_DIALDIGIT_LEN_MAX+1]

caller number,null terminated ASCII

6.3.2.5 char TelCallIncomingCallInfo_t::szCallingPartySubAddress[TAPI_CALL_SUBADDRESS_LEN_MAX+1]

called subaddress,null terminated ASCII

6.3.2.6 TelCallingNameInfo_t TelCallIncomingCallInfo_t::CallingNameInfo

Call name info. If there is no information from network, this information will be NULL.

6.3.2.7 TelCallUusInfo_t TelCallIncomingCallInfo_t::UusInfo

User -to User signaling information. (UUS1 only is supported). If there is no information from network, this information will be NULL.

6.3.2.8 TelCallAlertingPattern_t TelCallIncomingCallInfo_t::AlertingPattern

Alerting pattern

6.3.2.9 TelCallRedirectionInfo_t TelCallIncomingCallInfo_t::RedirectInfo

The data for the Call Redirect information. If there is no information from network, this information will be NULL.

6.3.2.10 int TelCallIncomingCallInfo_t::Signal

Signal Value received from the network (ref: 3gpp spec).

The documentation for this struct was generated from the following file:

- TelCall.h

6.4 TelCallingNameInfo_t Struct Reference

`#include <TelCall.h>`

6.4.1 Detailed Description

This structure defines the calling name information

Data Fields

- TelCallNameMode_t NameMode
- char szNameData [TAPI_CALLING_NAME_SIZE_MAX]

6.4.2 Field Documentation

6.4.2.1 TelCallNameMode_t TelCallingNameInfo_t::NameMode

Display mode of the szNameData.

6.4.2.2 char TelCallingNameInfo_t::szNameData[TAPI_CALLING_NAME_SIZE_MAX]

Calling party name string.

The documentation for this struct was generated from the following file:

- TelCall.h

6.5 TelCallRedirectionInfo_t Struct Reference

`#include <TelCall.h>`

6.5.1 Detailed Description

This structure contains information about call redirection details.

Data Fields

- char szRedirectedNumber [TAPI_CALL_DIALDIGIT_LEN_MAX+1]
- char szRedirectSubAddress [TAPI_CALL_SUBADDRESS_LEN_MAX+1]

6.5.2 Field Documentation

6.5.2.1 char TelCallRedirectionInfo_t::szRedirectedNumber[TAPI_CALL_DIALDIGIT_LEN_MAX+1]

Redirected party number

6.5.2.2 char TelCallRedirectionInfo_t::szRedirectSubAddress[TAPI_CALL_SUBADDRESS_LEN_MAX+1]

Redirect party sub-address

The documentation for this struct was generated from the following file:

- TelCall.h

6.6 TelCallSetupParams_t Struct Reference

```
#include <TelCall.h>
```

6.6.1 Detailed Description

This structure contains information about an outgoing call.

Data Fields

- TelCallType_t CallType
- char szNumber [TAPI_CALL_DIALDIGIT_LEN_MAX+1]
- char szCalledPartySubAddress [TAPI_CALL_SUBADDRESS_LEN_MAX+1]
- TelCallIdentityMode_t IdentityMode
- TelCallCugInfo_t pCugInfo
- TelCallUusInfo_t pUusInfo

6.6.2 Field Documentation

6.6.2.1 TelCallType_t TelCallSetupParams_t::CallType

sets type of call (voice, data, video, emergency)

6.6.2.2 char TelCallSetupParams_t::szNumber[TAPI_CALL_DIALDIGIT_LEN_MAX+1]

A string containing the destination phone number. This follows the dial number format.

6.6.2.3 char TelCallSetupParams_t::szCalledPartySubAddress[TAPI_CALL_SUBADDRESS_LEN_MAX+1]

A string containing the destination sub-address. This is Null terminated string. This is an optional parameter and it can be NULL.

6.6.2.4 TelCallIdentityMode_t TelCallSetupParams_t::IdentityMode

Sets Calling Line Identity mode. If default value TAPI_IDENTITY_DEFAULT is set, OEM will set the default value.

6.6.2.5 TelCallCugInfo_t TelCallSetupParams_t::pCugInfo

Closed User Group Info (Closed User Group Info).This is an optional parameter and it can be NULL.

6.6.2.6 TelCallUusInfo_t TelCallSetupParams_t::pUusInfo

User -to User signaling information. This is an optional parameter and it can be NULL.

The documentation for this struct was generated from the following file:

- TelCall.h

6.7 TelCallStatus_t Struct Reference

`#include <TelCall.h>`

6.7.1 Detailed Description

This structure contains information about call status.

Data Fields

- BOOL bMoCall
- char pNumber [TAPI_CALL_DIALDIGIT_LEN_MAX+1]
- TelCallType_t CallType
- TelCallActiveState_t CallState
- BOOL bConferenceState

6.7.2 Field Documentation

6.7.2.1 BOOL TelCallStatus_t::bMoCall

Status will be yes for MO Call. If MT call, then FALSE

6.7.2.2 char TelCallStatus_t::pNumber[TAPI_CALL_DIALDIGIT_LEN_MAX+1]

Mobile Number

6.7.2.3 TelCallType_t TelCallStatus_t::CallType

specifies type of call (voice, data, emergency)

6.7.2.4 TelCallActiveState_t TelCallStatus_t::CallState

Current Call state

6.7.2.5 BOOL TelCallStatus_t::bConferenceState

Whether Call is in Conference or not

The documentation for this struct was generated from the following file:

- TelCall.h

6.8 TelCallUserDataInfo_t Struct Reference

```
#include <TelCall.h>
```

6.8.1 Detailed Description

This structure contains information about user data information.

Data Fields

- unsigned int CallHandle
- TelCallUusInfo_t UusInfo

6.8.2 Field Documentation

6.8.2.1 unsigned int TelCallUserDataInfo_t::CallHandle

Call Handle

6.8.2.2 TelCallUusInfo_t TelCallUserDataInfo_t::UusInfo

User-User Information(UUS1) received from the network

The documentation for this struct was generated from the following file:

- TelCall.h

6.9 TelCallUusInfo_t Struct Reference

```
#include <TelCall.h>
```

6.9.1 Detailed Description

This structure defines the UUS information

Data Fields

- TelCallUusType_t UusType
- TelCallUusProtocolType_t ProtocolType
- BOOL bMoreDataPresent
- unsigned short UusDataLen
- unsigned char UusData [TAPI_CALL_UUS_DATA_LEN_MAX]

6.9.2 Field Documentation

6.9.2.1 TelCallUusType_t TelCallUusInfo_t::UusType

User-User type defined as per 3GPP 24.008.

6.9.2.2 TelCallUusProtocolType_t TelCallUusInfo_t::ProtocolType

User-User Protocol type defined as per 3GPP 24.008

6.9.2.3 BOOL TelCallUusInfo_t::bMoreDataPresent

Is more user-user information present

6.9.2.4 unsigned short TelCallUusInfo_t::UusDataLen

User-user data length.

6.9.2.5 unsigned char TelCallUusInfo_t::UusData[TAPI_CALL_UUS_DATA_LEN_MAX]

User-user data.

The documentation for this struct was generated from the following file:

- TelCall.h

6.10 TelNetworkConfigSettings_t Struct Reference

`#include <TelNetwork.h>`

6.10.1 Detailed Description

This structure has information related to network startup.

Data Fields

- BOOL bAutoSelection
- BOOL bPowerOnGprsAttach
- unsigned int NoOfNwModePresent
- TelNetworkMode_t NetworkOrder [TAPI_NETWORK_ORDER_MAX]
- TelNetworkServiceDomain_t ServiceDomain
- TelNetworkMode_t NetworkMode

6.10.2 Field Documentation

6.10.2.1 BOOL TelNetworkConfigSettings_t::bAutoSelection

Determines whether the Auto selection mode is enabled or not

6.10.2.2 BOOL TelNetworkConfigSettings_t::bPowerOnGprsAttach

Determines whether the Power on with GPRS attach is enabled or not

6.10.2.3 unsigned int TelNetworkConfigSettings_t::NoOfNwModePresent

No of network mode provided in the network_order array list

6.10.2.4 TelNetworkMode_t TelNetworkConfigSettings_t::NetworkOrder[TAPI_NETWORK_ORDER_MAX]

This is array of TAPI_ NETWORK_ORDER_MAX size of TelNetworkMode_t type

6.10.2.5 TelNetworkServiceDomain_t TelNetworkConfigSettings_t::ServiceDomain

This member sets the Service domain

6.10.2.6 TelNetworkMode_t TelNetworkConfigSettings_t::NetworkMode

This member sets the Network mode

The documentation for this struct was generated from the following file:

- TelNetwork.h

6.11 TelNetworkInfo_t Struct Reference

`#include <TelNetwork.h>`

6.11.1 Detailed Description

This structure has network information.

Data Fields

- TelNetworkServiceLevel_t ServiceLevel
- TelNetworkSystemId_t SysId
- BOOL bRoaming
- BOOL bDisplaySpn
- BOOL bDisplayRplmn

6.11.2 Field Documentation

6.11.2.1 TelNetworkServiceLevel_t TelNetworkInfo_t::ServiceLevel

Service level available(Full Service or Limited service etc)

6.11.2.2 TelNetworkSystemId_t TelNetworkInfo_t::SysId

System ID information

6.11.2.3 BOOL TelNetworkInfo_t::bRoaming

Roaming PLMN or not

6.11.2.4 BOOL TelNetworkInfo_t::bDisplaySpn

Service Provider Name should be displayed or not

6.11.2.5 BOOL TelNetworkInfo_t::bDisplayRplmn

Display of Registered PLMN required or not.

The documentation for this struct was generated from the following file:

- TelNetwork.h

6.12 TelNetworkList_t Struct Reference

`#include <TelNetwork.h>`

6.12.1 Detailed Description

This structure defines the network list.

Data Fields

- unsigned int NumOfNwAvailable
- TelNetworkSystemId_t SystemList [TAPI_NETWORK_LIST_MAX]

6.12.2 Field Documentation

6.12.2.1 unsigned int TelNetworkList_t::NumOfNwAvailable

Count for Network list

6.12.2.2 TelNetworkSystemId_t TelNetworkList_t::SystemList[TAPI_NETWORK_LIST_MAX]

List of Networks available

The documentation for this struct was generated from the following file:

- TelNetwork.h

6.13 TelNetworkRadioInfo_t Struct Reference

`#include <TelNetwork.h>`

6.13.1 Detailed Description

This structure defines the Radio information.

Data Fields

- unsigned short RxLevel
- unsigned short RxQuality
- unsigned short RssiBar

6.13.2 Field Documentation

6.13.2.1 unsigned short TelNetworkRadioInfo_t::RxLevel

Received radio signal level from the wireless stack.Radio signal strength indicator (RSSI) level.RSSI in positive dBm i.e. 75 means -75dBm.The valid range for AMPS:-110 to -89 dBm800 CDMA:-105 to -90 dBm1900 CDMA: -108 to -93 dBm GSM/WCDMA: -105 to -60 dBm

6.13.2.2 unsigned short TelNetworkRadioInfo_t::RxQuality

Radio signal quality received from the wireless stack.

6.13.2.3 unsigned short TelNetworkRadioInfo_t::RssiBar

Signal bars stating the signal strength

The documentation for this struct was generated from the following file:

- TelNetwork.h

6.14 TelNetworkSystemId_t Struct Reference

`#include <TelNetwork.h>`

6.14.1 Detailed Description

This structure defines the network system information.

Data Fields

- TelNetworkSystemType_t SystemType
- TelNetworkServiceType_t ServiceType
- TelNetworkSystemIdFormat_t SysIdFormat
- TelNetworkMode_t NetworkMode
- unsigned long SysId
- BOOL bPlmnForbidden
- BOOL bHome
- BOOL bEquivalentPlmn
- BOOL bNetworkIdentityAvail
- BOOL bNetworkTimeAvail
- BOOL bNetworkTimeZoneAvail
- BOOL bNetworkDlsAvail
- char szSystemName [TAPI_NETWORK_NAME_LEN_MAX]
- char szSpn [TAPI_NETWORK_SPN_LEN_MAX]
- TelNetworkTimeAndZone_t NetworkTimeZone
- BOOL bMncIncludesPcsDigit

6.14.2 Field Documentation

6.14.2.1 TelNetworkSystemType_t TelNetworkSystemId_t::SystemType

Network system technologies like GSM, CDMA, WCDMA, WLAN etc

6.14.2.2 TelNetworkServiceType_t TelNetworkSystemId_t::ServiceType

Network service type - Cs, Ps or combined

6.14.2.3 TelNetworkSystemIdFormat_t TelNetworkSystemId_t::SysIdFormat

SID or PLMN based on system type

6.14.2.4 TelNetworkMode_t TelNetworkSystemId_t::NetworkMode

Network mode Information

6.14.2.5 unsigned long TelNetworkSystemId_t::SysId

PLMN Information

6.14.2.6 BOOL TelNetworkSystemId_t::bPlmnForbidden

Forbidden PLMN or Not

6.14.2.7 BOOL TelNetworkSystemId_t::bHome

Home PLMN Or not

6.14.2.8 BOOL TelNetworkSystemId_t::bEquivalentPlmn

Equivalent PLMN Or not

6.14.2.9 BOOL TelNetworkSystemId_t::bNetworkIdentityAvail

Network Identity Available or not

6.14.2.10 BOOL TelNetworkSystemId_t::bNetworkTimeAvail

Network Time Available or not

6.14.2.11 BOOL TelNetworkSystemId_t::bNetworkTimeZoneAvail

Network Timezone Available or not

6.14.2.12 BOOL TelNetworkSystemId_t::bNetworkDlsAvail

Network DLS available or not

6.14.2.13 char TelNetworkSystemId_t::szSystemName[TAPI_NETWORK_NAME_LEN_MAX]

PLMN Name or Network Name (Null terminated string)

6.14.2.14 char TelNetworkSystemId_t::szSpn[TAPI_NETWORK_SPN_LEN_MAX]

Service Provider Name (Null Terminated string)

6.14.2.15 TelNetworkTimeAndZone_t TelNetworkSystemId_t::NetworkTimeZone

If NetwotkTimezone present, then timezone value

6.14.2.16 BOOL TelNetworkSystemId_t::bMncIncludesPcsDigit

To determine whether PLMN is US PLMN or not

The documentation for this struct was generated from the following file:

- TelNetwork.h

6.15 TelNetworkTimeAndZone_t Struct Reference

```
#include <TelNetwork.h>
```

6.15.1 Detailed Description

This structure defines the network time and zone information.

Data Fields

- unsigned short Year
- unsigned short Month
- unsigned short Day
- unsigned short Hour
- unsigned short Minute
- unsigned short Second
- unsigned short TimeZone
- TelNetworkDayLightSavingAdj_t DaylightSave

6.15.2 Field Documentation

6.15.2.1 unsigned short TelNetworkTimeAndZone_t::Year

Year information

6.15.2.2 unsigned short TelNetworkTimeAndZone_t::Month

Month information

6.15.2.3 unsigned short TelNetworkTimeAndZone_t::Day

Day

6.15.2.4 unsigned short TelNetworkTimeAndZone_t::Hour

Hour

6.15.2.5 unsigned short TelNetworkTimeAndZone_t::Minute

Minute

6.15.2.6 unsigned short TelNetworkTimeAndZone_t::Second

Second

6.15.2.7 unsigned short TelNetworkTimeAndZone_t::TimeZone

Network time zone

6.15.2.8 TelNetworkDayLightSavingAdj_t TelNetworkTimeAndZone_t::DaylightSave

day light saving adjustment time

The documentation for this struct was generated from the following file:

- TelNetwork.h

6.16 TelSimApnData_t Struct Reference

`#include <TelSim.h>`

6.16.1 Detailed Description

This structure defines the Access Point Name data

Data Fields

- unsigned long AclLen
- unsigned short Acl [TAPI_SIM_ACL_MAX_LEN]

6.16.2 Field Documentation

6.16.2.1 unsigned long TelSimApnData_t::AclLen

Acess point name control list length.

6.16.2.2 unsigned short TelSimApnData_t::Acl[TAPI_SIM_ACL_MAX_LEN]

Acl list

The documentation for this struct was generated from the following file:

- TelSim.h

6.17 TelSimCallbackEvent_t Struct Reference

`#include <TelSim.h>`

6.17.1 Detailed Description

This structure is used as a callback event notificaions in Telephony for applictions.

Data Fields

- int RequestId
- int EventStatus
- unsigned int DataLen
- unsigned char * pData
- TelTapiEventType_t EventType

6.17.2 Field Documentation

6.17.2.1 int TelSimCallbackEvent_t::RequestId

Handle to SIM -RequestId in LiMo

6.17.2.2 int TelSimCallbackEvent_t::EventStatus

SIM event status - Operation result or Access result depending on API

6.17.2.3 unsigned int TelSimCallbackEvent_t::DataLen

Data length

6.17.2.4 unsigned char* TelSimCallbackEvent_t::pData

Data associated with the event

6.17.2.5 TelTapiEventType_t TelSimCallbackEvent_t::EventType

Event type

The documentation for this struct was generated from the following file:

- TelSim.h

6.18 TelSimDeviceContext_t Struct Reference

`#include <TelSim.h>`

6.18.1 Detailed Description

This structure defines the SIM status data and SIM device context data

Data Fields

- TelSimStatusData_t SimStatusData
- TelSimDeviceContextData_t SimDeviceCtxData

6.18.2 Field Documentation

6.18.2.1 TelSimStatusData_t TelSimDeviceContext_t::SimStatusData

SIM data.

6.18.2.2 TelSimDeviceContextData_t TelSimDeviceContext_t::SimDeviceCtxData

SIM device context data.

The documentation for this struct was generated from the following file:

- TelSim.h

6.19 TelSimDeviceContextData_t Struct Reference

`#include <TelSim.h>`

6.19.1 Detailed Description

This structure defines the SIM device context data

Data Fields

- BOOL bIsDeviceCtxdataValid
- BOOL bBdnEnable
- BOOL bFdnEnable
- BOOL bAclEnable
- BOOL bImsiInvalidated
- unsigned long apnCount
- union {
  TelSimSstInfo_t SstInfo
  TelSimUstInfo_t UstInfo
  } SstData

- TelSimGroupIdentifierInfo_t Gid1Info
- TelSimGroupIdentifierInfo_t Gid2Info
- TelSimApnData_t ApnData

6.19.2 Field Documentation

6.19.2.1 BOOL TelSimDeviceContextData_t::bIsDeviceCtxdataValid

Indicates if device conext (this) data is valid.

6.19.2.2 BOOL TelSimDeviceContextData_t::bBdnEnable

BDN mode enabled or not.

6.19.2.3 BOOL TelSimDeviceContextData_t::bFdnEnable

FDN mode enabled or not.

6.19.2.4 BOOL TelSimDeviceContextData_t::bAclEnable

ACL(Access Point Name Control List) enabled or not.

6.19.2.5 BOOL TelSimDeviceContextData_t::bImsiInvalidated

IMSI Invalidated or not.

6.19.2.6 unsigned long TelSimDeviceContextData_t::apnCount

Number of Access point name.

6.19.2.7 TelSimSstInfo_t TelSimDeviceContextData_t::SstInfo

SIM Service table information.

6.19.2.8 TelSimUstInfo_t TelSimDeviceContextData_t::UstInfo

USIM Service table information.

6.19.2.9 union { ... } TelSimDeviceContextData_t::SstData

SS data information

6.19.2.10 TelSimGroupIdentifierInfo_t TelSimDeviceContextData_t::Gid1Info

Group Identifier 1.

6.19.2.11 TelSimGroupIdentifierInfo_t TelSimDeviceContextData_t::Gid2Info

Group Identifier 2.

6.19.2.12 TelSimApnData_t TelSimDeviceContextData_t::ApnData

Access point data.

The documentation for this struct was generated from the following file:

- TelSim.h

6.20 TelSimEccData_t Struct Reference

`#include <TelSim.h>`

6.20.1 Detailed Description

This data structure defines the data for the Ecc information.

Data Fields

- TelUsimEccInfo_t UeccInfo [TAPI_SIM_UECC_RECORD_COUNT_MAX]
- TelSimEccInfo_t EccInfo

6.20.2 Field Documentation

6.20.2.1 TelUsimEccInfo_t TelSimEccData_t::UeccInfo[TAPI_SIM_UECC_RECORD_COUNT_MAX]

Usim Emergency Call Code information structure

6.20.2.2 TelSimEccInfo_t TelSimEccData_t::EccInfo

Emergency Call Code information

The documentation for this struct was generated from the following file:

- TelSim.h

6.21 TelSimEccInfo_t Struct Reference

`#include <TelSim.h>`

6.21.1 Detailed Description

This enumeration defines the Ecc information.

Data Fields

- char szEcc1 [TAPI_SIM_ECC_CODE_LEN]
- char szEcc2 [TAPI_SIM_ECC_CODE_LEN]
- char szEcc3 [TAPI_SIM_ECC_CODE_LEN]
- char szEcc4 [TAPI_SIM_ECC_CODE_LEN]
- char szEcc5 [TAPI_SIM_ECC_CODE_LEN]

6.21.2 Field Documentation

6.21.2.1 char TelSimEccInfo_t::szEcc1[TAPI_SIM_ECC_CODE_LEN]

Emergency Call Code 1 info. ECC is coded in BCD format. Unused bytes is set to 0xFF.

6.21.2.2 char TelSimEccInfo_t::szEcc2[TAPI_SIM_ECC_CODE_LEN]

Emergency Call Code 2 info ECC is coded in BCD format. Unused bytes is set to 0xFF.

6.21.2.3 char TelSimEccInfo_t::szEcc3[TAPI_SIM_ECC_CODE_LEN]

Emergency Call Code 3 info ECC is coded in BCD format. Unused bytes is set to 0xFF.

6.21.2.4 char TelSimEccInfo_t::szEcc4[TAPI_SIM_ECC_CODE_LEN]

Emergency Call Code 4 info ECC is coded in BCD format. Unused bytes is set to 0xFF.

6.21.2.5 char TelSimEccInfo_t::szEcc5[TAPI_SIM_ECC_CODE_LEN]

Emergency Call Code 5 info ECC is coded in BCD format. Unused bytes is set to 0xFF.

The documentation for this struct was generated from the following file:

- TelSim.h

6.22 TelSimEvent_t Struct Reference

`#include <TelSim.h>`

6.22.1 Detailed Description

This structure is used as a callback event notifications for applications.

Data Fields

- int RequestId
- int EventStatus
- unsigned int DataLen
- unsigned char ∗ pData

6.22.2 Field Documentation

6.22.2.1 int TelSimEvent_t::RequestId

RequestId

6.22.2.2 int TelSimEvent_t::EventStatus

Sim Event Status

6.22.2.3 unsigned int TelSimEvent_t::DataLen

Data length

6.22.2.4 unsigned char∗ TelSimEvent_t::pData

Data associated with the event

The documentation for this struct was generated from the following file:

- TelSim.h

6.23 TelSimFileAccessGeneralResp_t Struct Reference

`#include <TelSim.h>`

6.23.1 Detailed Description

This data structure defines the response for SIM update file.

Data Fields

- TelSimFileID_t FileId
- TelSimAccessType_t AccessType

6.23.2 Field Documentation

6.23.2.1 TelSimFileID_t TelSimFileAccessGeneralResp_t::FileId

SIM file ID

6.23.2.2 TelSimAccessType_t TelSimFileAccessGeneralResp_t::AccessType

Access type

The documentation for this struct was generated from the following file:

- TelSim.h

6.24 TelSimFileID_t Struct Reference

`#include <TelSim.h>`

6.24.1 Detailed Description

This structure contains information about File name and path.

Data Fields

- TelSimFileName_t FileName
- unsigned char szPath [128]

6.24.2 Field Documentation

6.24.2.1 TelSimFileName_t TelSimFileID_t::FileName

Elementary FileName enumeration

6.24.2.2 unsigned char TelSimFileID_t::szPath[128]

Absolute szPath of the elementary file. [Null Terminated]

The documentation for this struct was generated from the following file:

- TelSim.h

6.25 TelSimFileInfo_t Struct Reference

`#include <TelSim.h>`

6.25.1 Detailed Description

This structure defines the SIM device Elementary file information read

Data Fields

- BOOL bValid
- TelSimFileID_t FileId
- TelSimFileType_t FileType
- TelSimAcl_t AcRead
- TelSimAcl_t AcUpdate
- TelSimAcl_t AcInvalidate
- TelSimAcl_t AcRehabilitate
- unsigned long RecordLen
- unsigned long RecordCount
- unsigned long FileSize

6.25.2 Field Documentation

6.25.2.1 BOOL TelSimFileInfo_t::bValid

Valid or not

6.25.2.2 TelSimFileID_t TelSimFileInfo_t::FileId

File identifier

6.25.2.3 TelSimFileType_t TelSimFileInfo_t::FileType

File type and structure

6.25.2.4 TelSimAcl_t TelSimFileInfo_t::AcRead

Access condition for read operation

6.25.2.5 TelSimAcl_t TelSimFileInfo_t::AcUpdate

Access condition for update operation

6.25.2.6 TelSimAcl_t TelSimFileInfo_t::AcInvalidate

Access condition for invalidate operation

6.25.2.7 TelSimAcl_t TelSimFileInfo_t::AcRehabilitate

Access condition for rehabilitate operation

6.25.2.8 unsigned long TelSimFileInfo_t::RecordLen

Length of one record in file

6.25.2.9 unsigned long TelSimFileInfo_t::RecordCount

Number of records in file

6.25.2.10 unsigned long TelSimFileInfo_t::FileSize

File size

The documentation for this struct was generated from the following file:

- TelSim.h

6.26 TelSimGroupIdentifierInfo_t Struct Reference

`#include <TelSim.h>`

6.26.1 Detailed Description

This data structure defines the data for the group identifier information.

Data Fields

- unsigned short GroupIdentifierLen
- char szGroupIdentifier [TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX]

6.26.2 Field Documentation

6.26.2.1 unsigned short TelSimGroupIdentifierInfo_t::GroupIdentifierLen

Group identifier length

6.26.2.2 char TelSimGroupIdentifierInfo_t::szGroupIdentifier[TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX]

Group identifier

The documentation for this struct was generated from the following file:

- TelSim.h

6.27 TelSimIccIdInfo_t Struct Reference

```
#include <TelSim.h>
```

6.27.1 Detailed Description

This data structure defines the data for the Iccid information.

Data Fields

- unsigned long IccidLength
- unsigned short Iccid [TAPI_SIM_ICCIDLENGTH]

6.27.2 Field Documentation

6.27.2.1 unsigned long TelSimIccIdInfo_t::IccidLength

Integrated Circuit Card id length

6.27.2.2 unsigned short TelSimIccIdInfo_t::Iccid[TAPI_SIM_ICCIDLENGTH]

Unique card identification number

The documentation for this struct was generated from the following file:

- TelSim.h

6.28 TelSimImsiInfo_t Struct Reference

`#include <TelSim.h>`

6.28.1 Detailed Description

This data structure defines the data for the Imsi information.

Data Fields

- BOOL bValid
- char szMcc [TAPI_SIM_MCC_CODE_LEN+1]
- char szMnc [TAPI_SIM_MNC_CODE_LEN+1]
- char szMsin [TAPI_SIM_MSIN_CODE_LEN+1]

6.28.2 Field Documentation

6.28.2.1 BOOL TelSimImsiInfo_t::bValid

Valid(ACTIVATE) IMSI (TRUE) or Deactivated (FALSE)

6.28.2.2 char TelSimImsiInfo_t::szMcc[TAPI_SIM_MCC_CODE_LEN+1]

mobile country code

6.28.2.3 char TelSimImsiInfo_t::szMnc[TAPI_SIM_MNC_CODE_LEN+1]

mobile network code

6.28.2.4 char TelSimImsiInfo_t::szMsin[TAPI_SIM_MSIN_CODE_LEN+1]

mobile station identification number

The documentation for this struct was generated from the following file:

- TelSim.h

6.29 TelSimLanguageInfo_t Struct Reference

`#include <TelSim.h>`

6.29.1 Detailed Description

This data structure defines the data for the Language information.

Data Fields

- unsigned long LpCount
- TelSimLangIndicationCode_t lp [TAPI_SIM_LI_MAX_LEN]

6.29.2 Field Documentation

6.29.2.1 unsigned long TelSimLanguageInfo_t::LpCount

Language preference count; zero indicates that the LP file is not available or it doesn't have any data.

6.29.2.2 TelSimLangIndicationCode_t TelSimLanguageInfo_t::lp[TAPI_SIM_LI_MAX_LEN]

Language preference code

The documentation for this struct was generated from the following file:

- TelSim.h

6.30 TelSimPb2gData_t Struct Reference

`#include <TelSim.h>`

6.30.1 Detailed Description

This data struture defines the 2G phone book data storage.

Data Fields

- unsigned short NumLen
- unsigned short NumType
- unsigned short NameLen
- unsigned short NameEncryptType
- unsigned short Number [TAPI_SIM_NUM_TEXT_MAX_LEN]
- unsigned short Name [TAPI_SIM_NUM_TEXT_MAX_LEN]

6.30.2 Field Documentation

6.30.2.1 unsigned short TelSimPb2gData_t::NumLen

Number length

6.30.2.2 unsigned short TelSimPb2gData_t::NumType

Number type [refer to TelSimTypeOfNum_t]

6.30.2.3 unsigned short TelSimPb2gData_t::NameLen

Name string length

6.30.2.4 unsigned short TelSimPb2gData_t::NameEncryptType

Name string encryption type [refer to TelSimTextEncrypt_t]

6.30.2.5 unsigned short TelSimPb2gData_t::Number[TAPI_SIM_NUM_TEXT_MAX_LEN]

Number data

6.30.2.6 unsigned short TelSimPb2gData_t::Name[TAPI_SIM_NUM_TEXT_MAX_LEN]

Name string

The documentation for this struct was generated from the following file:

- TelSim.h

6.31 TelSimPbEntryInfo_t Struct Reference

`#include <TelSim.h>`

6.31.1 Detailed Description

This data structure gives the phone book entry information.

Data Fields

- TelSimPbFileType_t StorageFileType
- unsigned short PbIndexMin
- unsigned short PbIndexMax
- unsigned short PbNumLenMax
- unsigned short PbTextLenMax

6.31.2 Field Documentation

6.31.2.1 TelSimPbFileType_t TelSimPbEntryInfo_t::StorageFileType

Storage file type

6.31.2.2 unsigned short TelSimPbEntryInfo_t::PbIndexMin

Phone book minimum index

6.31.2.3 unsigned short TelSimPbEntryInfo_t::PbIndexMax

Phone book maximum index

6.31.2.4 unsigned short TelSimPbEntryInfo_t::PbNumLenMax

Phone number maximum length

6.31.2.5 unsigned short TelSimPbEntryInfo_t::PbTextLenMax

Text maximum length

The documentation for this struct was generated from the following file:

- TelSim.h

6.32 TelSimPbReadResp_t Struct Reference

`#include <TelSim.h>`

6.32.1 Detailed Description

This data structure gives the phone book read response information.

Data Fields

- TelSimPbAccessResult_t PbAccessResult
- TelSimPbRecordData_t PbStorageData

6.32.2 Field Documentation

6.32.2.1 TelSimPbAccessResult_t TelSimPbReadResp_t::PbAccessResult

Indicates the result of phonebook update/read operation.

6.32.2.2 TelSimPbRecordData_t TelSimPbReadResp_t::PbStorageData

Data read from the phonebook file.

The documentation for this struct was generated from the following file:

- TelSim.h

6.33 TelSimPbRecordData_t Struct Reference

`#include <TelSim.h>`

6.33.1 Detailed Description

This structure defines the Phone book record data.

Data Fields

- TelSimPbFileType_t StorageFileType
- unsigned short Index
- TelSimPb2gData_t pPbData

6.33.2 Field Documentation

6.33.2.1 TelSimPbFileType_t TelSimPbRecordData_t::StorageFileType

Storage file type

6.33.2.2 unsigned short TelSimPbRecordData_t::Index

Index of the record being read/updated.

6.33.2.3 TelSimPb2gData_t TelSimPbRecordData_t::pPbData

Phone book contact info.

The documentation for this struct was generated from the following file:

- TelSim.h

6.34 TelSimPbStorageInfo_t Struct Reference

`#include <TelSim.h>`

6.34.1 Detailed Description

This data structure defines the phone book storage count information.

Data Fields

- TelSimPbFileType_t StorageFileType
- unsigned short TotalRecordCount
- unsigned short UsedRecordCount

6.34.2 Field Documentation

6.34.2.1 TelSimPbFileType_t TelSimPbStorageInfo_t::StorageFileType

Storage file type

6.34.2.2 unsigned short TelSimPbStorageInfo_t::TotalRecordCount

Total record count

6.34.2.3 unsigned short TelSimPbStorageInfo_t::UsedRecordCount

Used record count

The documentation for this struct was generated from the following file:

- TelSim.h

6.35 TelSimPbUpdateResp_t Struct Reference

`#include <TelSim.h>`

6.35.1 Detailed Description

This data structure gives the phone book update(add/edit/delete) response.

Data Fields

- TelSimPbAccessResult_t PbAccessResult
- TelSimPbStorageInfo_t PbStorageInfo
- unsigned short PbIndex

6.35.2 Field Documentation

6.35.2.1 TelSimPbAccessResult_t TelSimPbUpdateResp_t::PbAccessResult

Indicates the result of phonebook update/read operation.

6.35.2.2 TelSimPbStorageInfo_t TelSimPbUpdateResp_t::PbStorageInfo

Indicates the total and used phonebook entries for the updated storage type (ADN/FDN, etc).

6.35.2.3 unsigned short TelSimPbUpdateResp_t::PbIndex

Phone book record index that was updated/deleted.

The documentation for this struct was generated from the following file:

- TelSim.h

6.36 TelSimPinData_t Struct Reference

`#include <TelSim.h>`

6.36.1 Detailed Description

This structure contains information about pin data.

Data Fields

- TelSimPinType_t PinType
- char szPinCode [TAPI_SIM_PIN_LENGTH+1]

6.36.2 Field Documentation

6.36.2.1 TelSimPinType_t TelSimPinData_t::PinType

Pin type

6.36.2.2 char TelSimPinData_t::szPinCode[TAPI_SIM_PIN_LENGTH+1]

PIN code

The documentation for this struct was generated from the following file:

- TelSim.h

6.37 TelSimPinInfo_t Struct Reference

`#include <TelSim.h>`

6.37.1 Detailed Description

This data structure defines the data for the PIN Informtion.

Data Fields

- TelSimPinType_t PinType
- int RemainingAttempts

6.37.2 Field Documentation

6.37.2.1 TelSimPinType_t TelSimPinInfo_t::PinType

Specifies the PIN or PUK type.

6.37.2.2 int TelSimPinInfo_t::RemainingAttempts

Number of attempts remaining for PIN/PUK verification.

The documentation for this struct was generated from the following file:

- TelSim.h

6.38 TelSimReadFileData_t Struct Reference

`#include <TelSim.h>`

6.38.1 Detailed Description

This structure contains all read SIM elementary file information.

Data Fields

- TelSimFileID_t FileId
- unsigned short DataLen
- unsigned char Data

6.38.2 Field Documentation

6.38.2.1 TelSimFileID_t TelSimReadFileData_t::FileId

File Id

6.38.2.2 unsigned short TelSimReadFileData_t::DataLen

Data Length

6.38.2.3 unsigned char TelSimReadFileData_t::Data

SIM EF Data

The documentation for this struct was generated from the following file:

- TelSim.h

6.39 TelSimReadFileRequestData_t Struct Reference

`#include <TelSim.h>`

6.39.1 Detailed Description

This data structure defines the data for the Read file result information.

Data Fields

- TelSimFileID_t FileID
- TelSimFileType_t FileType
- unsigned long Offset
- unsigned long RecordIndex
- unsigned long RecordIndexFrom
- unsigned long RecordIndexTo
- TelSimRecordMode_t RecordMode
- unsigned long DataLength

6.39.2 Field Documentation

6.39.2.1 TelSimFileID_t TelSimReadFileRequestData_t::FileID

sim file id to read

6.39.2.2 TelSimFileType_t TelSimReadFileRequestData_t::FileType

sim file type

6.39.2.3 unsigned long TelSimReadFileRequestData_t::Offset

Offset if read binary

6.39.2.4 unsigned long TelSimReadFileRequestData_t::RecordIndex

record index if read record

6.39.2.5 unsigned long TelSimReadFileRequestData_t::RecordIndexFrom

start index from

6.39.2.6 unsigned long TelSimReadFileRequestData_t::RecordIndexTo

end index to

6.39.2.7 TelSimRecordMode_t TelSimReadFileRequestData_t::RecordMode

record mode if read record

6.39.2.8 unsigned long TelSimReadFileRequestData_t::DataLength

data length to read

The documentation for this struct was generated from the following file:

- TelSim.h

6.40 TelSimServiceTable_t Struct Reference

`#include <TelSim.h>`

6.40.1 Detailed Description

This data structure defines the data for the Sst (SIM service table) information.

Data Fields

- BOOL bPIN1DisableFunction
- BOOL bAdn
- BOOL bFdn
- BOOL bSms
- BOOL bAoc
- BOOL bCcp
- BOOL bPlmnSelector
- BOOL bRfu1
- BOOL bMsisdn
- BOOL bExtension1
- BOOL bExtension2
- BOOL bSmsParams
- BOOL blnd
- BOOL bCellBroadCastMsgId
- BOOL bGroupIdLevel1
- BOOL bGroupIdLevel2
- BOOL bSpn
- BOOL bSdn
- BOOL bExtension3
- BOOL bRfu2
- BOOL bVgcsGroupIdList
- BOOL bVbsGroupIDList
- BOOL bEnhancedMultilevelPrecedencePreemptionService
- BOOL bAutoAnswerForeMLPP
- BOOL bSmsCbdDataDownload
- BOOL bSmsPpDataDownload
- BOOL bMenuSelection
- BOOL bCallControl
- BOOL bProactiveSimCommand
- BOOL bCellBroadcastMsgIdRanges
- BOOL bBdn
- BOOL bExtension4
- BOOL bDePersonalizedControlKeys
- BOOL bCoOperativeNetworkList
- BOOL bSmsr
- BOOL bNetworkAlertIndInMS
- BOOL bMoSMSControl
- BOOL bGPRS
- BOOL bImage

- BOOL bSolsa
- BOOL bUSSDSupportedInCallControl
- BOOL bRunAtCommand
- BOOL bUserControlledPLMNSelectorAccessTech
- BOOL bOperatorControlledPlmnSelectorAccessTech
- BOOL bHplmnSelectorAccessTech
- BOOL bCPBCCHInfo
- BOOL bInvestigationScan
- BOOL bExtenedCapaConfigParams
- BOOL bMexe
- BOOL bRplmnLastUsedAccessTech

6.40.2 Field Documentation

6.40.2.1 BOOL TelSimServiceTable_t::bPIN1DisableFunction

PIN1 disable function

6.40.2.2 BOOL TelSimServiceTable_t::bAdn

abbreviated Dialing number

6.40.2.3 BOOL TelSimServiceTable_t::bFdn

fixed Dialing number

6.40.2.4 BOOL TelSimServiceTable_t::bSms

short message storage

6.40.2.5 BOOL TelSimServiceTable_t::bAoc

advice of charge

6.40.2.6 BOOL TelSimServiceTable_t::bCcp

capability configuration parameters

6.40.2.7 BOOL TelSimServiceTable_t::bPlmnSelector

plmn selector

6.40.2.8 BOOL TelSimServiceTable_t::bRfu1

rfu

6.40.2.9 BOOL TelSimServiceTable_t::bMsisdn

msisdn

6.40.2.10 BOOL TelSimServiceTable_t::bExtension1

extension1

6.40.2.11 BOOL TelSimServiceTable_t::bExtension2

extension2

6.40.2.12 BOOL TelSimServiceTable_t::bSmsParams

sms paramteres

6.40.2.13 BOOL TelSimServiceTable_t::blnd

last number dialled

6.40.2.14 BOOL TelSimServiceTable_t::bCellBroadCastMsgId

cell broadcast message identifier

6.40.2.15 BOOL TelSimServiceTable_t::bGroupIdLevel1

group identifier level 1

6.40.2.16 BOOL TelSimServiceTable_t::bGroupIdLevel2

group identifier level 2

6.40.2.17 BOOL TelSimServiceTable_t::bSpn

service provider name

6.40.2.18 BOOL TelSimServiceTable_t::bSdn

service Dialing number

6.40.2.19 BOOL TelSimServiceTable_t::bExtension3

extension3

6.40.2.20 BOOL TelSimServiceTable_t::bRfu2

rfu

6.40.2.21 BOOL TelSimServiceTable_t::bVgcsGroupIdList

vgcs group identifier (EF-VGCS, EF-VGCSS)

6.40.2.22 BOOL TelSimServiceTable_t::bVbsGroupIDList

vbs group identifier (EF-VBS, EF-VBSS)

6.40.2.23 BOOL TelSimServiceTable_t::bEnhancedMultilevelPrecedencePreemptionService

enhanced multi-level precedence and pre-emption service

6.40.2.24 BOOL TelSimServiceTable_t::bAutoAnswerForeMLPP

automatic answer fro eMLPP

6.40.2.25 BOOL TelSimServiceTable_t::bSmsCbdDataDownload

data download via sms-cb

6.40.2.26 BOOL TelSimServiceTable_t::bSmsPpDataDownload

data download via sms-pp

6.40.2.27 BOOL TelSimServiceTable_t::bMenuSelection

menu selection

6.40.2.28 BOOL TelSimServiceTable_t::bCallControl

call control

6.40.2.29 BOOL TelSimServiceTable_t::bProactiveSimCommand

proactive sim command

6.40.2.30 BOOL TelSimServiceTable_t::bCellBroadcastMsgIdRanges

cell broadcast message identifier ranges

6.40.2.31 BOOL TelSimServiceTable_t::bBdn

barred Dialing numbers

6.40.2.32 BOOL TelSimServiceTable_t::bExtension4

extension 4

6.40.2.33 BOOL TelSimServiceTable_t::bDePersonalizedControlKeys

de-personalization control keys

6.40.2.34 BOOL TelSimServiceTable_t::bCoOperativeNetworkList

co-operative network list

6.40.2.35 BOOL TelSimServiceTable_t::bSmsr

short message status reports

6.40.2.36 BOOL TelSimServiceTable_t::bNetworkAlertIndInMS

network's indication of alerting in the MS

6.40.2.37 BOOL TelSimServiceTable_t::bMoSMSControl

mobile-originated short message control by sim

6.40.2.38 BOOL TelSimServiceTable_t::bGPRS

gprs

6.40.2.39 BOOL TelSimServiceTable_t::bImage

image

6.40.2.40 BOOL TelSimServiceTable_t::bSolsa

support of local service area

6.40.2.41 BOOL TelSimServiceTable_t::bUSSDSupportedInCallControl

ussd string data object supported in call control

6.40.2.42 BOOL TelSimServiceTable_t::bRunAtCommand

RUN AT COMMAND command

6.40.2.43 BOOL TelSimServiceTable_t::bUserControlledPLMNSelectorAccessTech

user controlled PLMN selector with Access technology

6.40.2.44 BOOL TelSimServiceTable_t::bOperatorControlledPlmnSelectorAccessTech

operator controlled PLMN selector with Access technology

6.40.2.45 BOOL TelSimServiceTable_t::bHplmnSelectorAccessTech

HPLMN selector with access technology

6.40.2.46 BOOL TelSimServiceTable_t::bCPBCCHInfo

CPBCCH information

6.40.2.47 BOOL TelSimServiceTable_t::bInvestigationScan

investigation scan

6.40.2.48 BOOL TelSimServiceTable_t::bExtenedCapaConfigParams

extended capability configuration parameters

6.40.2.49 BOOL TelSimServiceTable_t::bMexe

MExE

6.40.2.50 BOOL TelSimServiceTable_t::bRplmnLastUsedAccessTech

RPLMN last used access technology

The documentation for this struct was generated from the following file:

- TelSim.h

6.41 TelSimSstInfo_t Struct Reference

`#include <TelSim.h>`

6.41.1 Detailed Description

This data structure defines the data for the Sim service table information.

Data Fields

- unsigned long SstLen
- unsigned short SstData [TAPI_SIM_SST_MAX_LEN]
- TelSimServiceTable_t Sst

6.41.2 Field Documentation

6.41.2.1 unsigned long TelSimSstInfo_t::SstLen

SIM service table length

6.41.2.2 unsigned short TelSimSstInfo_t::SstData[TAPI_SIM_SST_MAX_LEN]

SIM service table data

6.41.2.3 TelSimServiceTable_t TelSimSstInfo_t::Sst

This data structure defines the data for the Sst (SIM service table) information.

The documentation for this struct was generated from the following file:

- TelSim.h

6.42 TelSimStatusData_t Struct Reference

`#include <TelSim.h>`

6.42.1 Detailed Description

This data structure defines the SIM card status data.

Data Fields

- TelSimCardStatus_t CardStatus
- TelSimPinStatus_t UpinStatus
- TelSimPinStatus_t Pin1Status
- TelSimPinStatus_t Pin2Status
- TelSimCardType_t CardType
- unsigned short EccRecordCount
- TelSimEccData_t EccRecord
- TelSimLanguageInfo_t LpInfo
- unsigned long IccidLen
- unsigned short Iccid [TAPI_SIM_ICCIDLENGTH]

6.42.2 Field Documentation

6.42.2.1 TelSimCardStatus_t TelSimStatusData_t::CardStatus

Provides the card status

6.42.2.2 TelSimPinStatus_t TelSimStatusData_t::UpinStatus

Universal PIN status

6.42.2.3 TelSimPinStatus_t TelSimStatusData_t::Pin1Status

Provides the Pin 1 status

6.42.2.4 TelSimPinStatus_t TelSimStatusData_t::Pin2Status

Provides the Pin 2 status

6.42.2.5 TelSimCardType_t TelSimStatusData_t::CardType

Provides the SIM card type

6.42.2.6 unsigned short TelSimStatusData_t::EccRecordCount

Provides the Emergency Call Code record count

6.42.2.7 TelSimEccData_t TelSimStatusData_t::EccRecord

Provides the Emergency call codes

6.42.2.8 TelSimLanguageInfo_t TelSimStatusData_t::LpInfo

Provides the Language preference information

6.42.2.9 unsigned long TelSimStatusData_t::IccidLen

Provides the Integrated Circuit Card id length

6.42.2.10 unsigned short TelSimStatusData_t::Iccid[TAPI_SIM_ICCIDLENGTH]

Provides the Integrated Circuit Card id

The documentation for this struct was generated from the following file:

- TelSim.h

6.43 TelSimUpdateFileRequestData_t Struct Reference

`#include <TelSim.h>`

6.43.1 Detailed Description

This structure contains the information required to update elementary files in SIM.

Data Fields

- TelSimFileID_t FileId
- TelSimFileType_t FileType
- unsigned long Offset
- unsigned long RecordIndex
- TelSimRecordMode_t RecordMode
- unsigned long DataLen
- unsigned char ∗ szData

6.43.2 Field Documentation

6.43.2.1 TelSimFileID_t TelSimUpdateFileRequestData_t::FileId

sim file id to update

6.43.2.2 TelSimFileType_t TelSimUpdateFileRequestData_t::FileType

sim file type

6.43.2.3 unsigned long TelSimUpdateFileRequestData_t::Offset

Offset, if update for a binary file

6.43.2.4 unsigned long TelSimUpdateFileRequestData_t::RecordIndex

Record index, if update is for record type file

6.43.2.5 TelSimRecordMode_t TelSimUpdateFileRequestData_t::RecordMode

Record index, if update is for record type file

6.43.2.6 unsigned long TelSimUpdateFileRequestData_t::DataLen

szData length to update

6.43.2.7 unsigned char∗ TelSimUpdateFileRequestData_t::szData

SIM EF Data

The documentation for this struct was generated from the following file:

- TelSim.h

6.44 TelSimUstInfo_t Struct Reference

```
#include <TelSim.h>
```

6.44.1 Detailed Description

This data structure defines the data for the USim service table information.

Data Fields

- unsigned long UsimSstLen
- unsigned short UstData [TAPI_SIM_SST_MAX_LEN]
- TelUSimServiceTable_t UsimSst

6.44.2 Field Documentation

6.44.2.1 unsigned long TelSimUstInfo_t::UsimSstLen

USIM service table length

6.44.2.2 unsigned short TelSimUstInfo_t::UstData[TAPI_SIM_SST_MAX_LEN]

USIM service table data - Complete service table from sim.

6.44.2.3 TelUSimServiceTable_t TelSimUstInfo_t::UsimSst

This data structure defines the data for the UST (USIM service table) information.

The documentation for this struct was generated from the following file:

- TelSim.h

6.45 TelSmsAddressInfo_t Struct Reference

`#include <TelNetText.h>`

6.45.1 Detailed Description

This structure defines the properties of a dialing number Type of Number, Numbering Plan Indicator, length and the actual number.

Data Fields

- unsigned int DialNumLen
- TelSimTypeOfNum_t Ton
- TelSimNumberingPlanIdentity_t Npi
- unsigned char szDiallingNum [TAPI_SIM_SMSP_ADDRESS_LEN+1]

6.45.2 Field Documentation

6.45.2.1 unsigned int TelSmsAddressInfo_t::DialNumLen

length of address

6.45.2.2 TelSimTypeOfNum_t TelSmsAddressInfo_t::Ton

Type of number

6.45.2.3 TelSimNumberingPlanIdentity_t TelSmsAddressInfo_t::Npi

Numbering plan identification

6.45.2.4 unsigned char TelSmsAddressInfo_t::szDiallingNum[TAPI_SIM_SMSP_ADDRESS_LEN+1]

destination address

The documentation for this struct was generated from the following file:

- TelNetText.h

6.46 TelSmsCbConfig_t Struct Reference

`#include <TelNetText.h>`

6.46.1 Detailed Description

This structure defines the different parameters of CB configuration

Data Fields

- BOOL bCBEnabled
- unsigned char SelectedId
- int MsgIdCount
- unsigned short MsgIDs [TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX]

6.46.2 Field Documentation

6.46.2.1 BOOL TelSmsCbConfig_t::bCBEnabled

CB service state. If cb_enabled is true then cell broadcast service will be enabled and underlying modem will enable CB Channel to receiving CB messages. Otherwise CB service will be disabled, underlying modem will deactivate the CB channel. (enabled/disabled)

6.46.2.2 unsigned char TelSmsCbConfig_t::SelectedId

CBMI Identifier selected (all or some)

6.46.2.3 int TelSmsCbConfig_t::MsgIdCount

CB message ID count

6.46.2.4 unsigned short TelSmsCbConfig_t::MsgIDs[TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX]

CB message ID information

The documentation for this struct was generated from the following file:

- TelNetText.h

6.47 TelSmsCbMsg_t Struct Reference

`#include <TelNetText.h>`

6.47.1 Detailed Description

This structure defines a cell broadcast message.

Data Fields

- TelSmsCbMsgType_t CbMsgType
- unsigned short Length
- char szMsgData [TAPI_NETTEXT_CB_PAGE_SIZE_MAX+1]

6.47.2 Field Documentation

6.47.2.1 TelSmsCbMsgType_t TelSmsCbMsg_t::CbMsgType

Cell Broadcast message type

6.47.2.2 unsigned short TelSmsCbMsg_t::Length

Size of array szMsgData (which is actual TPDU message)

6.47.2.3 char TelSmsCbMsg_t::szMsgData[TAPI_NETTEXT_CB_PAGE_SIZE_MAX+1]

Cell broadcast message data[Refer 3GPP TS 23.041 9.4.1]

The documentation for this struct was generated from the following file:

- TelNetText.h

6.48 TelSmsData_t Struct Reference

`#include <TelNetText.h>`

6.48.1 Detailed Description

This structure defines the data Related to SimIndex,Message Status and SMS Data Stored.

Data Fields

- signed short SimIndex
- TelSmsMsgStatus_t MsgStatus
- TelSmsDatapackageInfo_t SmsData

6.48.2 Field Documentation

6.48.2.1 signed short TelSmsData_t::SimIndex

Index where SMS is stored. If index is invalid (-1), then it means the message is not stored in SIM.

6.48.2.2 TelSmsMsgStatus_t TelSmsData_t::MsgStatus

Message status. If index is invalid, Msg status shall be ignored.

6.48.2.3 TelSmsDatapackageInfo_t TelSmsData_t::SmsData

SMS message

The documentation for this struct was generated from the following file:

- TelNetText.h

6.49 TelSmsDatapackageInfo_t Struct Reference

`#include <TelNetText.h>`

6.49.1 Detailed Description

This structure defines the fields related to an Sms like sim index, TPDU and length

Data Fields

- char * pSCA
- int MsgLength
- char szData [TAPI_NETTEXT_SMDATA_SIZE_MAX+1]

6.49.2 Field Documentation

6.49.2.1 char* TelSmsDatapackageInfo_t::pSCA

Service Centre address - an optional parameter. If this parameter is not present, then this field will be Null.If present, the valid service centre address information is filled as per TelSmsAddressInfo_t and provided as char

6.49.2.2 int TelSmsDatapackageInfo_t::MsgLength

Size of array szData (which is actual TPDU message)

6.49.2.3 char TelSmsDatapackageInfo_t::szData[TAPI_NETTEXT_SMDATA_SIZE_MAX+1]

SMS TPDU message

The documentation for this struct was generated from the following file:

- TelNetText.h

6.50 TelSmsParams_t Struct Reference

`#include <TelNetText.h>`

6.50.1 Detailed Description

This structure defines different fields involved in setting the parameters of a particular sms in EFsmsp.

Data Fields

- unsigned char RecordIndex
- unsigned char RecordLen
- unsigned long AlphaIdLen
- char szAlphaId [TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX+1]
- BYTE ParamIndicator
- TelSmsAddressInfo_t TpDestAddr
- TelSmsAddressInfo_t TpSvcCntrAddr
- unsigned short TpProtocolId
- unsigned short TpDataCodingScheme
- unsigned short TpValidityPeriod

6.50.2 Field Documentation

6.50.2.1 unsigned char TelSmsParams_t::RecordIndex

Index

6.50.2.2 unsigned char TelSmsParams_t::RecordLen

SMS Parameter Record length

6.50.2.3 unsigned long TelSmsParams_t::AlphaIdLen

Alpha id length

6.50.2.4 char TelSmsParams_t::szAlphaId[TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX+1]

Alpha id .It is a unique identifier for each row/record in EFsmsp

6.50.2.5 BYTE TelSmsParams_t::ParamIndicator

SMS parameter indicator is a 1 byte value. Each bit in this value indicates the presence/absence of the sms header parameters.If the parameter is present the corresponding bit value is set to 0.If the parameter is absent then it is set as 1.Refer 3GPP TS 31.02 :4.4.23

6.50.2.6 TelSmsAddressInfo_t TelSmsParams_t::TpDestAddr

TP-destination address (TP-DA)

6.50.2.7 TelSmsAddressInfo_t TelSmsParams_t::TpSvcCntrAddr

TP-service center address

6.50.2.8 unsigned short TelSmsParams_t::TpProtocolId

TP-protocol Id

6.50.2.9 unsigned short TelSmsParams_t::TpDataCodingScheme

TP-data coding scheme

6.50.2.10 unsigned short TelSmsParams_t::TpValidityPeriod

TP-validity period

The documentation for this struct was generated from the following file:

- TelNetText.h

6.51 TelSmsStoredMsgCountInfo_t Struct Reference

`#include <TelNetText.h>`

6.51.1 Detailed Description

This structure defines the different parameters that are related to the message count in a particular memory(Phone/SIM)

Data Fields

- unsigned int TotalCount
- int UsedCount
- int IndexList [TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX]

6.51.2 Field Documentation

6.51.2.1 unsigned int TelSmsStoredMsgCountInfo_t::TotalCount

Total count of messages stored in SIM

6.51.2.2 int TelSmsStoredMsgCountInfo_t::UsedCount

Stored message count in SIM in SIM

6.51.2.3 int TelSmsStoredMsgCountInfo_t::IndexList[TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX]

Message index list stored in SIM. And the maximum size of this array can be of totalCount.This array contains the list of index values in which the messages are stored.i.e. index_list[totalcount] = [2,3] denotes that indexs 2 and 3 are stored(valid) and others not(empty).

The documentation for this struct was generated from the following file:

- TelNetText.h

6.52 TelSsCallBarringInfo_t Struct Reference

```
#include <TelSs.h>
```

6.52.1 Detailed Description

This structure defines the different parameters related to call barring.

Data Fields

- TelSsCallBarOperationMode_t Mode
- TelSsCallBarType_t Type
- TelSsCallType_t CallType
- char szPassword [TAPI_SS_GSM_BARR_PW_LEN_MAX]

6.52.2 Field Documentation

6.52.2.1 TelSsCallBarOperationMode_t TelSsCallBarringInfo_t::Mode

Barring mode

6.52.2.2 TelSsCallBarType_t TelSsCallBarringInfo_t::Type

Barring type

6.52.2.3 TelSsCallType_t TelSsCallBarringInfo_t::CallType

Call type

6.52.2.4 char TelSsCallBarringInfo_t::szPassword[TAPI_SS_GSM_BARR_PW_LEN_MAX]

password

The documentation for this struct was generated from the following file:

- TelSs.h

6.53 TelSsCallBarringRecord_t Struct Reference

`#include <TelSs.h>`

6.53.1 Detailed Description

This structure defines the values for ss call barring record

Data Fields

- TelSsStatus_t Status
- TelSsCallBarType_t Flavour
- TelSsTeleService_t TeleCommService

6.53.2 Field Documentation

6.53.2.1 TelSsStatus_t TelSsCallBarringRecord_t::Status

Ss status

6.53.2.2 TelSsCallBarType_t TelSsCallBarringRecord_t::Flavour

Call barring types providing various barring conditions on that basis call be barred

6.53.2.3 TelSsTeleService_t TelSsCallBarringRecord_t::TeleCommService

Tele service

The documentation for this struct was generated from the following file:

- TelSs.h

6.54 TelSsCallWaitingRecord_t Struct Reference

```
#include <TelSs.h>
```

6.54.1 Detailed Description

This structure defines the values for ss call waiting record

Data Fields

- TelSsStatus_t Status
- TelSsTeleService_t TeleCommService

6.54.2 Field Documentation

6.54.2.1 TelSsStatus_t TelSsCallWaitingRecord_t::Status

SS status

6.54.2.2 TelSsTeleService_t TelSsCallWaitingRecord_t::TeleCommService

Tele service

The documentation for this struct was generated from the following file:

- TelSs.h

6.55 TelSsCapabilityInfo_t Struct Reference

```
#include <TelSs.h>
```

6.55.1 Detailed Description

This structure defines the various parameters in a capability info type

Data Fields

- BOOL bCFSupported
- BOOL bCBSupported
- BOOL bMPTYSupported
- BOOL bCWSupported
- BOOL bCallHoldSupported
- BOOL bECTSupported

6.55.2 Field Documentation

6.55.2.1 BOOL TelSsCapabilityInfo_t::bCFSupported

TRUE will indicate call forwarding supported, FALSE otherwise.

6.55.2.2 BOOL TelSsCapabilityInfo_t::bCBSupported

TRUE will indicate call barring supported, FALSE otherwise.

6.55.2.3 BOOL TelSsCapabilityInfo_t::bMPTYSupported

TRUE will indicate Multi Party call handling supported, FALSE otherwise

6.55.2.4 BOOL TelSsCapabilityInfo_t::bCWSupported

TRUE will indicate call waiting supported, FALSE otherwise

6.55.2.5 BOOL TelSsCapabilityInfo_t::bCallHoldSupported

TRUE will indicate call hold supported, FALSE otherwise.

6.55.2.6 BOOL TelSsCapabilityInfo_t::bECTSupported

TRUE will indicate explicit call transfer supported, FALSE otherwise.

The documentation for this struct was generated from the following file:

- TelSs.h

6.56 TelSsCliRecord_t Struct Reference

`#include <TelSs.h>`

6.56.1 Detailed Description

This structure defines the values for Calling line identity service

Data Fields

- TelSsLineIdentificationType_t IdentificationType
- TelSsCliStatus_t CliStatus

6.56.2 Field Documentation

6.56.2.1 TelSsLineIdentificationType_t TelSsCliRecord_t::IdentificationType

Various line identification types

6.56.2.2 TelSsCliStatus_t TelSsCliRecord_t::CliStatus

Line indentification status from network

The documentation for this struct was generated from the following file:

- TelSs.h

6.57 TelSsForwardInfo_t Struct Reference

```
#include <TelSs.h>
```

6.57.1 Detailed Description

This structure defines the different parameters related to forward info.

Data Fields

- TelSsForwardMode_t Mode
- TelSsForwardType_t Type
- TelSsForwardWhen_t Condition
- char szPhoneNumber [TAPI_CALL_DIALDIGIT_LEN_MAX]

6.57.2 Field Documentation

6.57.2.1 TelSsForwardMode_t TelSsForwardInfo_t::Mode

Forward Mode

6.57.2.2 TelSsForwardType_t TelSsForwardInfo_t::Type

Forward Type

6.57.2.3 TelSsForwardWhen_t TelSsForwardInfo_t::Condition

Forward Condition

6.57.2.4 char TelSsForwardInfo_t::szPhoneNumber[TAPI_CALL_DIALDIGIT_LEN_MAX]

Phone Number

The documentation for this struct was generated from the following file:

- TelSs.h

6.58 TelSsForwardRecord_t Struct Reference

```
#include <TelSs.h>
```

6.58.1 Detailed Description

This structure defines the values for ss call forwarding record

Data Fields

- TelSsStatus_t Status
- TelSsForwardWhen_t ForwardCondition
- TelSsTeleService_t TeleCommService
- BOOL bCallForwardingNumberPresent
- unsigned char szCallForwardingNumber [TAPI_CALL_DIALDIGIT_LEN_MAX]

6.58.2 Field Documentation

6.58.2.1 TelSsStatus_t TelSsForwardRecord_t::Status

Call forwarding SS status

6.58.2.2 TelSsForwardWhen_t TelSsForwardRecord_t::ForwardCondition

Call forward types providing various conditions when call can be forwarded

6.58.2.3 TelSsTeleService_t TelSsForwardRecord_t::TeleCommService

Tele service type

6.58.2.4 BOOL TelSsForwardRecord_t::bCallForwardingNumberPresent

Call forwarding number present or not

6.58.2.5 unsigned char TelSsForwardRecord_t::szCallForwardingNumber[TAPI_CALL_DIALDIGIT_LEN_MAX]

forwarded number.[Null Terminated string]

The documentation for this struct was generated from the following file:

- TelSs.h

6.59 TelSsInfo_t Struct Reference

`#include <TelSs.h>`

6.59.1 Detailed Description

related all kind of SS information

Data Fields

- TelSsType_t SsType
- TelSsStatus_t SsStatus
- UINT NumberOfRecords
- TelSsRecord_t SsRecord
- TelSsTeleService_t SsTeleserviceType

6.59.2 Field Documentation

6.59.2.1 TelSsType_t TelSsInfo_t::SsType

Supplementary service type

6.59.2.2 TelSsStatus_t TelSsInfo_t::SsStatus

SS status information

6.59.2.3 UINT TelSsInfo_t::NumberOfRecords

Number of records present for Call Barring / Call forwarding / Call waiting. In case of any error, this parameter value will be set to zero.

6.59.2.4 TelSsRecord_t TelSsInfo_t::SsRecord

SS record (Union of Call Barring / Call Forwarding/Call waiting / line identification information). This structure can be Null when number_of_records parameter is zero.

6.59.2.5 TelSsTeleService_t TelSsInfo_t::SsTeleserviceType

Tele service type

The documentation for this struct was generated from the following file:

- TelSs.h

6.60 TelSsRecord_t Union Reference

`#include <TelSs.h>`

6.60.1 Detailed Description

This structure defines the parameters for SS record type

Data Fields

- TelSsCallWaitingRecord_t WaitingRecord
- TelSsCallBarringRecord_t BarringRecord
- TelSsCliRecord_t CliRecord
- TelSsForwardRecord_t ForwardingRecord

6.60.2 Field Documentation

6.60.2.1 TelSsCallWaitingRecord_t TelSsRecord_t::WaitingRecord

Call waiting information

6.60.2.2 TelSsCallBarringRecord_t TelSsRecord_t::BarringRecord

Call barring information

6.60.2.3 TelSsCliRecord_t TelSsRecord_t::CliRecord

Line identification information

6.60.2.4 TelSsForwardRecord_t TelSsRecord_t::ForwardingRecord

Call forward record information

The documentation for this union was generated from the following file:

- TelSs.h

6.61 TelSsUssdMsgInfo_t Struct Reference

`#include <TelSs.h>`

6.61.1 Detailed Description

This structure defines the values for USSD request type.

Data Fields

- TelSsUssdIndType_t IndType
- UINT8 Dcs
- int UssdStringLength
- char szUssdString [TAPI_SS_USSD_DATA_SIZE_MAX]

6.61.2 Field Documentation

6.61.2.1 TelSsUssdIndType_t TelSsUssdMsgInfo_t::IndType

USSD Indication type (This field is used in case of sending USSD Response only)

6.61.2.2 UINT8 TelSsUssdMsgInfo_t::Dcs

Data coding scheme value as per 3GPP TS 23.038

6.61.2.3 int TelSsUssdMsgInfo_t::UssdStringLength

USSD data length

6.61.2.4 char TelSsUssdMsgInfo_t::szUssdString[TAPI_SS_USSD_DATA_SIZE_MAX]

USSD data

The documentation for this struct was generated from the following file:

- TelSs.h

6.62 TelTapiErrorInfo_t Struct Reference

`#include <TelUtility.h>`

6.62.1 Detailed Description

This structure holds information like length and actual error data.

Data Fields

- unsigned int Length
- void ∗ pErrorData

6.62.2 Field Documentation

6.62.2.1 unsigned int TelTapiErrorInfo_t::Length

Length of error data

6.62.2.2 void∗ TelTapiErrorInfo_t::pErrorData

Error information exchanged between OEM and application(Not Null terminated string). For Debugging/Display purpose on screen.

The documentation for this struct was generated from the following file:

- TelUtility.h

6.63 TelTapiEvent_t Struct Reference

`#include <TapiCommon.h>`

6.63.1 Detailed Description

This structure defines the Telephony Event

Data Fields

- TelTapiEventClass_t EventClass
- TelTapiEventType_t EventType
- int RequestId
- int Status
- void ∗ pData

6.63.2 Field Documentation

6.63.2.1 TelTapiEventClass_t TelTapiEvent_t::EventClass

Event type

6.63.2.2 TelTapiEventType_t TelTapiEvent_t::EventType

Event type

6.63.2.3 int TelTapiEvent_t::RequestId

This parameter is applicable only for Asynchronous solicited responses. The request id for a service request is sent back in the asynchronous response. For unsolicited indications, request id will be -1

6.63.2.4 int TelTapiEvent_t::Status

This parameter is applicable only for Asynchronous solicited responses. This status parameter informs the actual result of Asynchronous response. For unsolicited indications, status will be -1

6.63.2.5 void∗ TelTapiEvent_t::pData

Event Data - Based on Event type response/indications. Event data information is captured in each API Section

The documentation for this struct was generated from the following file:

- TapiCommon.h

6.64 TelTapiTunnelInfo_t Struct Reference

`#include <TelUtility.h>`

6.64.1 Detailed Description

This structure holds information like length and actual tunneling data.

Data Fields

- unsigned int TunnelDataLen
- void ∗ pTunnelData

6.64.2 Field Documentation

6.64.2.1 unsigned int TelTapiTunnelInfo_t::TunnelDataLen

Length of tunnel data

6.64.2.2 void∗ TelTapiTunnelInfo_t::pTunnelData

Tunneling information exchanged between OEM and application(Not Null terminated string)

The documentation for this struct was generated from the following file:

- TelUtility.h

6.65 TelTapiVersionInfo_t Struct Reference

`#include <TelUtility.h>`

6.65.1 Detailed Description

This structure holds Telephony version information.

Data Fields

- int MajorVersion
- int MinorVersion

6.65.2 Field Documentation

6.65.2.1 int TelTapiVersionInfo_t::MajorVersion

Major Telephony version

6.65.2.2 int TelTapiVersionInfo_t::MinorVersion

Minor Telephony version

The documentation for this struct was generated from the following file:

- TelUtility.h

6.66 TelUsimEccInfo_t Struct Reference

`#include <TelSim.h>`

6.66.1 Detailed Description

This data structure defines the data for the USim emergency call code information.

Data Fields

- BOOL bUsed
- char szEcc [TAPI_SIM_ECC_CODE_LEN]
- unsigned short EccLen
- char szEccAlphaIdentifier [TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX]
- TelSimEccEmergencyServiceInfo_t EccEmergencyServiceInfo

6.66.2 Field Documentation

6.66.2.1 BOOL TelUsimEccInfo_t::bUsed

Userd or not

6.66.2.2 char TelUsimEccInfo_t::szEcc[TAPI_SIM_ECC_CODE_LEN]

Emergency Call Code info-ECC is coded in BCD format. Unused bytes is set to 0xFF.

6.66.2.3 unsigned short TelUsimEccInfo_t::EccLen

Ecc length

6.66.2.4 char TelUsimEccInfo_t::szEccAlphaIdentifier[TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX]

Alphabet identifer

6.66.2.5 TelSimEccEmergencyServiceInfo_t TelUsimEccInfo_t::EccEmergencyServiceInfo

ECC emergency service information

The documentation for this struct was generated from the following file:

- TelSim.h

6.67 TelUSimServiceTable_t Struct Reference

`#include <TelSim.h>`

6.67.1 Detailed Description

This data structure defines the data for the UST (USIM service table) information.

Data Fields

- BOOL bLocalPhoneBook
- BOOL bFdn
- BOOL bExtension2
- BOOL bSdn
- BOOL bExtension3
- BOOL bBdn
- BOOL bExtension4
- BOOL bOCIOCT
- BOOL bIciIct
- BOOL bSms
- BOOL bSmsr
- BOOL bSMSP
- BOOL bAoc
- BOOL bCcp
- BOOL bCellBroadcastMsgId
- BOOL bCellBroadcastMsgIdRanges
- BOOL bGroupIdLevel1
- BOOL bGroupIdLevel2
- BOOL bSpn
- BOOL bUserControlledPLMNSelectorAccessTech
- BOOL bMsisdn
- BOOL bImage
- BOOL bSolsa
- BOOL bEnhancedMultilevelPrecedencePreemptionService
- BOOL bAutoAnswerForeMLPP
- BOOL bRfu1
- BOOL bGsmAccess
- BOOL bSmsPpDataDownload
- BOOL bSmsCbdDataDownload
- BOOL bUsimCallControl
- BOOL bUsimMoSmsControl
- BOOL bRunAtCommand
- BOOL bShallBeSetToOne
- BOOL bEnabledServiceTable
- BOOL bAcl
- BOOL bDePersonalizedControlKeys
- BOOL bCoOperativeNetworkList
- BOOL bGsmSecurityContext
- BOOL bCpbcchiInformation

- BOOL bInvestigationScan
- BOOL bMexe
- BOOL bOperatorControlledPlmnSelectorAccessTech
- BOOL bHplmnSelectorAccessTech
- BOOL bExtension5
- BOOL bPlmnNetworkName
- BOOL bOperatorPlmnList
- BOOL bMailBoxDiallingNumbers
- BOOL bMsgWaitingIndStatus
- BOOL bCallForwardingIndicationStatus
- BOOL bRplmnLastUsedAccessTech
- BOOL bServiceProviderDisplayInfo
- BOOL bMms
- BOOL bExtension8
- BOOL bUsimGpsCallControl
- BOOL bMmsUserConnectivityParameters
- BOOL bNia
- BOOL bVgcsIdList
- BOOL bVbsGroupId

6.67.2 Field Documentation

6.67.2.1 BOOL TelUSimServiceTable_t::bLocalPhoneBook

local phone book

6.67.2.2 BOOL TelUSimServiceTable_t::bFdn

fixed Dialing number

6.67.2.3 BOOL TelUSimServiceTable_t::bExtension2

extension 2

6.67.2.4 BOOL TelUSimServiceTable_t::bSdn

service Dialing number

6.67.2.5 BOOL TelUSimServiceTable_t::bExtension3

extension 3

6.67.2.6 BOOL TelUSimServiceTable_t::bBdn

barred Dialing numbers

6.67.2.7 BOOL TelUSimServiceTable_t::bExtension4

extension 4

6.67.2.8 BOOL TelUSimServiceTable_t::bOCIOCT

outgoing call information

6.67.2.9 BOOL TelUSimServiceTable_t::bIciIct

incoming call information

6.67.2.10 BOOL TelUSimServiceTable_t::bSms

short message storage

6.67.2.11 BOOL TelUSimServiceTable_t::bSmsr

short message status reports

6.67.2.12 BOOL TelUSimServiceTable_t::bSMSP

short message service parameters

6.67.2.13 BOOL TelUSimServiceTable_t::bAoc

advice of charage

6.67.2.14 BOOL TelUSimServiceTable_t::bCcp

capability config parameters

6.67.2.15 BOOL TelUSimServiceTable_t::bCellBroadcastMsgId

cell broadcast message identifier

6.67.2.16 BOOL TelUSimServiceTable_t::bCellBroadcastMsgIdRanges

cell broadcast message identifier randges

6.67.2.17 BOOL TelUSimServiceTable_t::bGroupIdLevel1

group identifier level 1

6.67.2.18 BOOL TelUSimServiceTable_t::bGroupIdLevel2

group identifier level 2

6.67.2.19 BOOL TelUSimServiceTable_t::bSpn

service provide name

6.67.2.20 BOOL TelUSimServiceTable_t::bUserControlledPLMNSelectorAccessTech

user controlled plmn selector with access technology

6.67.2.21 BOOL TelUSimServiceTable_t::bMsisdn

msisdn

6.67.2.22 BOOL TelUSimServiceTable_t::bImage

image

6.67.2.23 BOOL TelUSimServiceTable_t::bSolsa

Solsa

6.67.2.24 BOOL TelUSimServiceTable_t::bEnhancedMultilevelPrecedencePreemptionService

enhanced multi-level precedence and pre-emption service

6.67.2.25 BOOL TelUSimServiceTable_t::bAutoAnswerForeMLPP

automatic answer fro eMLPP

6.67.2.26 BOOL TelUSimServiceTable_t::bRfu1

Rfu1

6.67.2.27 BOOL TelUSimServiceTable_t::bGsmAccess

gsm access

6.67.2.28 BOOL TelUSimServiceTable_t::bSmsPpDataDownload

data download via sms-pp

6.67.2.29 BOOL TelUSimServiceTable_t::bSmsCbdDataDownload

data download via sms-cb

6.67.2.30 BOOL TelUSimServiceTable_t::bUsimCallControl

usim call control

6.67.2.31 BOOL TelUSimServiceTable_t::bUsimMoSmsControl

mo-sms control by usim

6.67.2.32 BOOL TelUSimServiceTable_t::bRunAtCommand

RUN AT COMMAND command

6.67.2.33 BOOL TelUSimServiceTable_t::bShallBeSetToOne

shall be set to 1

6.67.2.34 BOOL TelUSimServiceTable_t::bEnabledServiceTable

enabled service table

6.67.2.35 BOOL TelUSimServiceTable_t::bAcl

APN control list

6.67.2.36 BOOL TelUSimServiceTable_t::bDePersonalizedControlKeys

de-personalization control keys

6.67.2.37 BOOL TelUSimServiceTable_t::bCoOperativeNetworkList

co-operative network list

6.67.2.38 BOOL TelUSimServiceTable_t::bGsmSecurityContext

gsm security context

6.67.2.39 BOOL TelUSimServiceTable_t::bCpbcchiInformation

cpbcch information

6.67.2.40 BOOL TelUSimServiceTable_t::bInvestigationScan

investigation scan

6.67.2.41 BOOL TelUSimServiceTable_t::bMexe

mexe

6.67.2.42 BOOL TelUSimServiceTable_t::bOperatorControlledPlmnSelectorAccessTech

Operator controlled PLMN selector with Access Technology

6.67.2.43 BOOL TelUSimServiceTable_t::bHplmnSelectorAccessTech

HPLMN selector with access technology

6.67.2.44 BOOL TelUSimServiceTable_t::bExtension5

extension 5

6.67.2.45 BOOL TelUSimServiceTable_t::bPlmnNetworkName

plmn network name

6.67.2.46 BOOL TelUSimServiceTable_t::bOperatorPlmnList

operator plmn list

6.67.2.47 BOOL TelUSimServiceTable_t::bMailBoxDiallingNumbers

mailbox Dialing numbers

6.67.2.48 BOOL TelUSimServiceTable_t::bMsgWaitingIndStatus

message waiting indication status

6.67.2.49 BOOL TelUSimServiceTable_t::bCallForwardingIndicationStatus

call forwarding indication status

6.67.2.50 BOOL TelUSimServiceTable_t::bRplmnLastUsedAccessTech

RPLMN last used access technology

6.67.2.51 BOOL TelUSimServiceTable_t::bServiceProviderDisplayInfo

service provider display information

6.67.2.52 BOOL TelUSimServiceTable_t::bMms

multimedia messaging service

6.67.2.53 BOOL TelUSimServiceTable_t::bExtension8

extension 8

6.67.2.54 BOOL TelUSimServiceTable_t::bUsimGpsCallControl

call control on gprs by usim

6.67.2.55 BOOL TelUSimServiceTable_t::bMmsUserConnectivityParameters

mms user connectivity parameters

6.67.2.56 BOOL TelUSimServiceTable_t::bNia

Network's indication of alerting in the MS (NIA)

6.67.2.57 BOOL TelUSimServiceTable_t::bVgcsIdList

VGCS Group Identifier List (EFVGCS and EFVGCSS)

6.67.2.58 BOOL TelUSimServiceTable_t::bVbsGroupId

VBS Group Identifier List (EFVBS and EFVBSS)

The documentation for this struct was generated from the following file:

- TelSim.h

Chapter 7

File Documentation

7.1 ComponentOverview.h File Reference

7.1.1 Detailed Description

Component Programming Interface Overview Header Template for Doxygen.

This file contains configuration information for documentation generation; it does not contain any API-specific information.

7.2 ITapiCall.h File Reference

7.2.1 Detailed Description

This file serves as a "C" header file defines functions for Tapi Call Services.

It contains a sample set of function prototypes that would be required by applications.

Note: Call related supplementary services are dependent on Operator/Market/Vendor. If any of the Supplementary Service

is not supported in underlying telephony provider due to the dependencies, then Call Related SS APIs may return

an error TAPI_API_NOT_SUPPORTED.

==

`#include "ITapiSs.h"`

`#include "TapiCommon.h"`

`#include "TelCall.h"`

Functions

- TapiResult_t TelTapiCallSetup (const TelCallSetupParams_t ∗pParams, unsigned int ∗pCallHandle, int ∗pRequestId)

  This function originates MO call setup. This is an asynchronous function. The function completes immediately and call setup proceeds in background.

- TapiResult_t TelTapiCallAnswer (unsigned int CallHandle, TelCallAnswerType_t AnsType, int ∗pRequestId)

  This function is used to answer the incoming call by accepting or rejecting the call.

- TapiResult_t TelTapiCallRelease (unsigned int CallHandle, int ∗pRequestId)

  This function releases the call identified by Call Handle irrespective of call is hold or active state.

- TapiResult_t TelTapiCallReleaseAll (int ∗pRequestId)

  This function releases the all the calls irrespective of the call is hold or active state. Due to this API request, lower layers will release the call and END Indication Event will be sent to all the Applications.

- TapiResult_t TelTapiCallSendDTMF (const char ∗pDtmfString, int ∗pRequestId)

  This function sends one or more DTMF digits during the call.

- TapiResult_t TelTapiCallRetrieve (unsigned int CallHandle, int ∗pRequestId)

  This function retrieves the held call. The call identified by Call Handle must be in held state.

- TapiResult_t TelTapiCallHold (unsigned int CallHandle, int ∗pRequestId)

  This function puts the given call on hold. The call identified by Call Handle should be in active state.

- TapiResult_t TelTapiCallJoin (unsigned int CallHandle1, unsigned int CallHandle2, int ∗pRequestId)

  This function joins the given two calls (one call in active conversation state and other call in held state) into conference.

- TapiResult_t TelTapiCallSplit (unsigned int CallHandle, int ∗pRequestId)

  This function triggers to splits a private call from multiparty call. This API allows creating a private communication with one of remote parties in a multiparty session. TAPI client application has to pass the call handle which needs to split from the multiparty call.

- TapiResult_t TelTapiCallExplicitTransfer (unsigned int CallHandle, int ∗pRequestId)

  This function triggers to do an explicit call transfer by connecting the two parties where in one party being active (active state) and another party being held (held state).

- TapiResult_t TelTapiCallGetStatus (unsigned int CallHandle, TelCallStatus_t ∗pCallStatus)

  This function gets status for the current call identified by Call Handle whenever application wants the call status. Call handle must be valid.

- TapiResult_t TelTapiCallGetDuration (unsigned int CallHandle, unsigned int ∗pDurationInSecs)

  This function gets duration of the given call. This is a synchronous function. Duration is accounted from the moment the connection is established, i.e. call goes into active state for first time.

- TapiResult_t TelTapiCallGetConferenceList (unsigned int CallHandle, unsigned int ∗pCallList, int ∗pNoOfCalls)

  This is a synchronous function returns all call handles within the given conference call.

7.3 ITapiNetText.h File Reference

7.3.1 Detailed Description

This file serves as a "C" header file defines functions for Tapi NetText Services.

It contains a sample set of function prototypes that would be required by applications.

Telephony provides SMS TPDU interface as defined by 3GPP TS 23.040 Rel 5 version for sending /receiving of SMS PDUs

directly from/to SMS Plug-in (Messaging Framework). Telephony provides various APIs to read/write/delete SMS from

SIM and provides functionality of SMS storage/retrieval related to SIM. Telephony provides interfaces only for

accessing SMS SIM memory not the Phone (ME) memory.

Based on SMS sending/receiving, Telephony FW shall just pass through the SMS PDU received from the Messaging FW to

the OEM and vice-versa. Segmentation/Reassembling of SMS PDUs and handling of User Data header and User data is

the functionality of SMS Plug-in in Messaging FW and shall not be functionality of TAPI library and Telephony Server.

==

```
#include "ITapiSim.h"

#include "tapi_utility.h"

#include "TapiCommon.h"

#include "TelNetText.h"
```

Functions

- TapiResult_t TelTapiSmsSend (const TelSmsDatapackageInfo_t ∗pDataPackage, BOOL bMoreMsgToSend, int ∗pRequestId)

  This API is used to send an SMS message to the network. This API interface allows transmitting SMS PDU as defined by 3GPP TS 23.040 (SMS-SUBMIT, SMS-COMMAND) to lower layers. This API allows keeping the dedicated link at lower layers by passing information as more messages to send in parameters. This will enable not to release if a dedicated connection is used for transmission.

- TapiResult_t TelTapiSmsRead (int Index, int ∗pRequestId)

  This function reads a message from SIM card storage. SIM index (index used when storing the SMS) number to be passed to in order to read particular SMS message. Total number of messages can be retrieved by TelTapiSmsGetCount function.

- TapiResult_t TelTapiSmsWrite (TelSmsMsgStatus_t MsgStatus, const TelSmsDatapackageInfo_t ∗pDataPackage, int ∗pRequestId)

  This function writes a message to SIM storage area.

- TapiResult_t TelTapiSmsDelete (signed short Index, int ∗pRequestId)

  This function deletes a message specified by index (SIM index used when storing the sms) number from the SIM storage area.

- TapiResult_t TelTapiSmsGetCount (int ∗pRequestId)

  This function is used to retrieve message count information like Total record count, used record count and the indices list for SIM Storage. This is an asynchronous function and the response will be sent back by event notification.

- TapiResult_t TelTapiSmsCBSetConfig (const TelSmsCbConfig_t ∗pCBConfig, int ∗pRequestId)

  This function is used to set SMS CB Message Identifiers in the appropriate EF-CBMI/EF-CBMIR file in (U)SIM. This function also indicates underlying provider to enable or disable CB Channel on which CB messages are received.

- TapiResult_t TelTapiSmsCBGetConfig (int ∗pRequestId)

  This function is used to retrieve SMS CB configuration options from EFcbmi file in the SIM.

- TapiResult_t TelTapiSmsSetSCA (const TelSmsAddressInfo_t ∗pSCA, int ∗pRequestId)

  This function is used to set SMS Service Centre Address information in order to send the SMS.

- TapiResult_t TelTapiSmsGetSCA (TelSmsAddressInfo_t ∗pSCA)

  This function is used to get current SMS Service Centre Address information.

- TapiResult_t TelTapiSmsSetPreferredBearer (TelSmsBearerType_t BearerType, int ∗pRequestId)

  This function is used to set SMS preferred bearer on which SMS has to be transmitted.

- TapiResult_t TelTapiSmsGetPreferredBearer (int ∗pRequestId)

  This function is used to get SMS preferred bearer on which SMS has to be transmitted. This is an asynchronous function.

- TapiResult_t TelTapiSmsGetParameters (const int SmsRecordIndex, int ∗pRequestId)

  This function is used to get the SMS parameters for a particular SMS (sent/received) based on the SIM index where it is stored. This is an asynchronous function.

- TapiResult_t TelTapiSmsSetParameters (const TelSmsParams_t SmsSetParameters, int ∗pRequestId)

  This function is used to set the short message service header parameters which are used in the origination of MO messages.

- TapiResult_t TelTapiSmsSendDeliveryReport (const TelSmsDatapackageInfo_t ∗pDataPackage, int ∗pRequestId)

  This function is used to send a delivery report for the received incoming SMS to the network.

7.4 ITapiNetwork.h File Reference

7.4.1 Detailed Description

This file serves as a "C" header file defines functions for Tapi Network.

It contains a sample set of function prototypes that would be required by applications.

==

`#include "TapiCommon.h"`

`#include "TelNetwork.h"`

Functions

- TapiResult_t TelTapiNwSetConfiguration (const TelNetworkConfigSettings_t ∗pConfigNetworkInfo, int ∗pRequestId)

  This API allows to set the network configuration settings at OEM based on which network registration will happen.

- TapiResult_t TelTapiNwSelectNetwork (const TelNetworkSystemId_t ∗pSystemId, int ∗pRequestId)

  This function requests the lower layers to select the network (PLMN) which has been selected by the user from the Network List displayed to the User.

- TapiResult_t TelTapiNwCancelManualSearch (int ∗pRequestId)

  This function is called to cancel the triggered manual network search.

- TapiResult_t TelTapiNwCancelManualSelection (int ∗pRequestId)

  This function is called to cancel manual network selection where user has already selected a network from the network list.

- TapiResult_t TelTapiNwGetNetworkInfo (int ∗pRequestId)

  This function provides an option to get the current network registration status information.

- TapiResult_t TelTapiNwSearchNetwork (int ∗pRequestId)

  This function sends a request to do manual network selection to search for the available networks and provide the Network List to the User/Application.

- TapiResult_t TelTapiNwSetSelectionMode (TelNetworkSelectionMode_t NwSelectionMode, int ∗pRequestId)

  This function is called when user does Automatic or Manual Network Selection. Request for Automatic NW selection or Manual NW selection will allow OEM provider to do PLMN Selection based on network selection preference.

- TapiResult_t TelTapiNwGetSelectionMode (int ∗pRequestId)

  This function requests for the present network selection mode i.e. automatic or manual.

- TapiResult_t TelTapiNwSetServiceDomain (TelNetworkServiceDomain_t ServiceDomain, int ∗pRequestId)

  This function is called when User/application wants to configure the service domain to only CS or Only PS or Both. This API triggers the underlying protocol stack to do register with Network for only CS services or only PS services or both based on the option set using this API.

- TapiResult_t TelTapiNwSetNetworkOrder (TelNetworkMode_t ∗pPrefNwOrder, int ∗pRequestId)

  This API allows lower layers to scan based on preferred network order based options passed.

- TapiResult_t TelTapiNwSetNetworkMode (TelNetworkMode_t NwMode, int ∗pRequestId)

  This function allows to configure the Network Mode (Automatic, GSM, WCDMA) at OEM provider based on which lower layers can scan only those radio access technologies for registering with network.

- TapiResult_t TelTapiNwSetNetworkBand (TelNetworkBandPreferred_t BandMode, TelNetworkBand_t Band, int ∗pRequestId)

  This function is invoked to set the network band and allows the underlying OEM provider to scan the set band.

7.5 ITapiSim.h File Reference

7.5.1 Detailed Description

This file serves as a "C" header file defines functions for Tapi SIM Services.

It contains a sample set of function prototypes that would be required by applications.

Note: Phonebook APIs are 2G SIM Specific in R1 timeframe. Some Operator may have support only 3G Phonebook.

In this case, 2G SIM Phonebook APIs may return TAPI_API_NOT_SUPPORTED as an error value in TapiResult_t.

==

`#include "limo_defines.h"`

`#include "TapiCommon.h"`

`#include "TelSim.h"`

Functions

- TapiResult_t TelTapiSimGetFileInfo (TelSimFileID_t FileId, int ∗pRequestId)

  This function is passed to OEM provider in order to fetch the Elementary File info. This functions fetches all the Information like Whether EF is present or not, If Present, no of records in EF and EF Data length etc. Application can make use this API to know the EF details prior calling SIM Read API.

- TapiResult_t TelTapiSimReadFile (const TelSimReadFileRequestData_t ∗pReadFileReqData, int ∗pRequestId)

  This function is passed to OEM provider in order to read an elementary file.

- TapiResult_t TelTapiSimUpdateFile (const TelSimUpdateFileRequestData_t ∗pUpdateFileReqData, int ∗pRequestId)

  This function is passed to OEM provider in-order to update records in an elementary File.

- TapiResult_t TelTapiSimVerifyPIN (const TelSimPinData_t ∗pPinData, int ∗pRequestId)

  This function performs PIN verification. This function performs PIN verification based on the pin type passed along with pin data.

- TapiResult_t TelTapiSimChangePIN (TelSimPinType_t PinType, const char ∗pOldPinData, const char ∗pNewPinData, int ∗pRequestId)

  This function changes PIN code based on the pin type passed along with old pin data and new pin data.

- TapiResult_t TelTapiSimDisablePIN (const TelSimPinData_t ∗pPinData, int ∗pRequestId)

  This function disables the PIN based on the Pin type and pin data passed from Application.

- TapiResult_t TelTapiSimEnablePIN (const TelSimPinData_t ∗pPinData, int ∗pRequestId)

  This function enables the PIN based on the Pin type and pin data passed from the Application.

- TapiResult_t TelTapiSimUnblockPIN (const TelSimPinData_t ∗pPukPinData, const TelSimPinData_t ∗pNewPinData, int ∗pRequestId)

  This function performs unblock PIN operation based on PUK information passed along with unblock information entered by the user.

- TapiResult_t TelTapiSimDisableFDN (const char ∗pPinData, int ∗pRequestId)

  This function disables FDN.

- TapiResult_t TelTapiSimEnableFDN (const char ∗pPinData, int ∗pRequestId)

  This function enables FDN.

- TapiResult_t TelTapiSimGetDevCtxdata (TelSimDeviceContext_t ∗pSimData)

  This function is used to get device context data of SIM. SIM device context data such as PIN Enabled status, FDN, BDN Status, IMSI information, ECC, LP, (U)SIM service table information, group identifier information will be passed.

- TapiResult_t TelTapiSimGetIccIdInfo (TelSimIccIdInfo_t ∗pIccIdInfo)

  This function gets ICC id information.

- TapiResult_t TelTapiSimGetPINStatus (TelSimPinStatus_t ∗pPinStatus, TelSimPinType_t ∗pPinType)

  This function gets the current PIN (PIN1 or UPIN) status.

- TapiResult_t TelTapiSimGetPIN2Status (TelSimPinStatus_t ∗pPin2Status)

  This function gets the current Pin2 status.

- TapiResult_t TelTapiSimGetFDNStatus (BOOL ∗pFdnStatus)

  This function is a synchronous API gets the current FDN status.

- TapiResult_t TelTapiSimGetSstInfo (TelSimSstInfo_t ∗pSimSstInfo)

  This function gets the SIM service table information.

- TapiResult_t TelTapiSimGetUstInfo (TelSimUstInfo_t ∗pSimUstInfo)

  This function gets the USIM service table information.

- TapiResult_t TelTapiSimGetImsiInfo (TelSimImsiInfo_t ∗pSimImsiInfo)

  This function is a synchronous API gets the IMSI information.

- TapiResult_t TelTapiSimGetEccInfo (TelSimEccInfo_t ∗pSimEccInfo, int ∗pCount)

  This function is a synchronous API used to get ECC data.

- TapiResult_t TelTapiSimGetLanguagePrefInfo (TelSimLanguageInfo_t ∗pSimLpInfo)

  This API is used to get the language preference information.

- TapiResult_t TelTapiSimGetLanguageIndInfo (TelSimLanguageInfo_t ∗pSimLiInfo)

  This API is used to get the language indication information.

- TapiResult_t TelTapiSimGetCardType (TelSimCardType_t ∗pCardInfo)

  This function is a synchronous API which gets the Card Type whether SIM Card is a GSM SIM or a 3G SIM.

- TapiResult_t TelTapiSimPBGetStorageCount (TelSimPbFileType_t StorageFileType, int ∗pRequestId)

  This API is used to get number of used records and total records in SIM phone book files. This API is applicable only to GSM phonebook (ADN and EXT1), FDN, SDN, LND and MSISDN.

- TapiResult_t TelTapiSimPBGetEntryInfo (TelSimPbFileType_t StorageFileType, int ∗pRequestId)

  This API is used to get max text length and max number length supported by SIM phone book elementary file.

- TapiResult_t TelTapiSimPBReadRecord (TelSimPbFileType_t StorageFileType, unsigned short Index, int ∗pRequestId)

  This API is used to read SIM phone book entry information from given storage type.

- TapiResult_t TelTapiSimPBUpdateRecord (const TelSimPbRecordData_t ∗pPbRecordData, int ∗pRequestId)

  This API is used to add or edit SIM phone book record entry information.

- TapiResult_t TelTapiSimPBDeleteRecord (TelSimPbFileType_t StorageFileType, unsigned short RecordIndex, int ∗pRequestId)

  This API is used to delete a SIM phone book record.

7.6 ITapiSs.h File Reference

7.6.1 Detailed Description

This file serves as a "C" header file defines functions for Tapi Supplementary Services.

It contains a sample set of function prototypes that would be required by applications.

Note: Supplementary services are dependent on Operator/Market/Vendor. If any of the Supplementary Service is

not supported in underlying telephony provider due to the dependencies, then SS APIs may return an error

TAPI_API_NOT_SUPPORTED.

==

```
#include "limo_defines.h"

#include "TapiCommon.h"

#include "TelSs.h"
```

Functions

- TapiResult_t TelTapiSsSetCallBarring (const TelSsCallBarringInfo_t ∗pBarringInfo, int ∗pRequestId)

  This function sends a request to activates/deactivates call barring. This service makes it possible for a mobile subscriber to activate barring of certain categories of outgoing or incoming calls according to a barring program which is selected from a set of one or more barring programs chosen at provision time and is valid for all outgoing or incoming calls, or just those associated with a specific basic service group. Also this API allows deactivating call barring based on categories set.

- TapiResult_t TelTapiSsGetCallBarringStatus (TelSsCallBarType_t BarType, TelSsCallType_t CallType, int ∗pRequestId)

  This function gets call bar status. The mobile subscriber may determine by subscription of a set of one or more unique barring programs what kind of outgoing or incoming calls should be barred.

- TapiResult_t TelTapiSsChangeBarPassword (const char ∗pOldPassword, const char ∗pNewPassword, int ∗pRequestId)

  This function allows changing of the barring password in Network.

- TapiResult_t TelTapiSsSetCallForward (const TelSsForwardInfo_t ∗pForwardInfo, int ∗pRequestId)

  This function allows to set (register/erase/activate/deactivate) call forwarding option at the Network. This enables to forward the calls to another party. This service permits a called mobile subscriber to have the network send all incoming calls, or just those associated with a specific Basic service group, addressed to the called mobile subscriber's directory number and which meet various call FORWARD condition to another directory number. The ability of the served mobile subscriber to originate calls is unaffected. If the service is activated, a call is forwarded only if under this call forward condition. Application has to call this API in order to explicitly register/erase/activate/deactivate the call forwarding. Request will be passed irrespective of whether already another SS call forward is ongoing.

- TapiResult_t TelTapiSsGetCallForwardStatus (TelSsForwardType_t Type, TelSsForwardWhen_t Condition, int ∗pRequestId)

  This function provides option to get the call forwarding status of different calls from Network.

- TapiResult_t TelTapiSsSetCallWaiting (BOOL bCwStatus, int ∗pRequestId)

  This function activates/deactivates the call waiting service. The call waiting service permits a mobile to be notified of an incoming call (as per basic call procedures) whilst the traffic channel (BM or LM) is not available for the incoming call and the mobile subscriber is engaged in an active or held call. Subsequently, the subscriber can accept, reject, or ignore the incoming call.

- TapiResult_t TelTapiSsGetCallWaitingStatus (int ∗pRequestId)

  This function queries the status of call waiting service. The call waiting service permits a mobile to be notified of an incoming call (as per basic call procedures) whilst the traffic channel (BM or LM) is not available for the incoming call and the mobile subscriber is engaged in an active or held call. Subsequently, the subscriber can accept, reject, or ignore the incoming call.

- TapiResult_t TelTapiSsVerifyBarPwdResponse (int ∗pRequestId, const char ∗pBarPwdInfo)

  This API allows Application to provide barring password requested by network for Call Barring Request or Change barring password request.

- TapiResult_t TelTapiSsSetRestrictCallId (TelSsRestrictIDStatus_t Status)

  This function sets the status of restrict call ID. After set, the phone will show/hide the caller number to the called party for all subsequent calls, regardless of type of call. This API assumes user service subscription allows changing this setting, sometimes referred to as 'temporary mode'. If the presentation mode is not temporary, the caller of this API will receive an error result.

- TapiResult_t TelTapiSsGetRestrictCallId (TelSsRestrictIDStatus_t ∗pStatus)

  This function gets the status of call ID restriction for all subsequent calls.

- TapiResult_t TelTapiSsQueryCapabilityInfo (int ∗pRequestId)

  This API allows Application to query what all SS services are supported at OEM.

- TapiResult_t TelTapiUssdSendRequest (const TelSsUssdMsgInfo_t ∗pAppMsg, int ∗pRequestId)

  This function is used to send USSD to the Network. The response to the request is a displayable string which will be returned to the Application via LiMo EventDelivery.

- TapiResult_t TelTapiUssdSendUserResponse (const TelSsUssdMsgInfo_t ∗pAppMsg, int ∗pRequestId)

  This function is called to send a USSD response provided by the User to the Network for the USSD indication received. This contains USSD response type, USSD Indication type and USSD Response data.

7.7 TapiCommon.h File Reference

7.7.1 Detailed Description

This file serves as a "C" header file defines functions for Tapi Common Interfaces like Callback Registration / De-registration.

It contains a sample set of function prototypes that would be required by applications.

==

```
#include "TelUtility.h"
```

Data Structures

- struct TelTapiEvent_t

Typedefs

- typedef bool(∗) TelAppCallback (const TelTapiEvent_t ∗pTapiEvent)

Enumerations

- enum TelTapiEventClass_t {

  TAPI_EVENT_NONE = -1, TAPI_EVENT_CLASS_CALL_VOICE = 0x00, TAPI_EVENT_CLASS_CALL_DATA, TAPI_EVENT_CLASS_NETTEXT,

  TAPI_EVENT_CLASS_NETWORK, TAPI_EVENT_CLASS_SS, TAPI_EVENT_CLASS_SIM, TAPI_EVENT_CLASS_ERR,

  TAPI_EVENT_CLASS_UTIL, TAPI_EVENT_CLASS_ALL }
- enum TelTapiEventType_t {

  TAPI_EVENT_CALL_SETUP_CNF = 1, TAPI_EVENT_CALL_ALERT_IND, TAPI_EVENT_CALL_CONNECTING_IND, TAPI_EVENT_CALL_CONNECTED_IND,

  TAPI_EVENT_CALL_INCOM_IND, TAPI_EVENT_CALL_ANSWER_CNF, TAPI_EVENT_CALL_RETRIEVE_CNF, TAPI_EVENT_CALL_RETRIEVE_IND,

  TAPI_EVENT_CALL_HOLD_IND, TAPI_EVENT_CALL_HOLD_CNF, TAPI_EVENT_CALL_TRANSFER_CNF, TAPI_EVENT_CALL_TRANSFER_IND,

  TAPI_EVENT_CALL_SETUPCONFERENCE_CNF, TAPI_EVENT_CALL_SETUPCONFERENCE_IND, TAPI_EVENT_CALL_SPLITCONFERENCE_CNF, TAPI_EVENT_CALL_SEND_DTMF_CNF,

  TAPI_EVENT_CALL_WAITING_IND, TAPI_EVENT_CALL_FORWARD_IND, TAPI_EVENT_CALL_USERINFO_IND, TAPI_EVENT_CALL_RELEASE_CNF,

  TAPI_EVENT_CALL_RELEASE_ALL_CNF, TAPI_EVENT_CALL_END_IND, TAPI_EVENT_NETTEXT_READ_SMS_CNF, TAPI_EVENT_NETTEXT_GET_COUNT_CNF,

  TAPI_EVENT_NETTEXT_SENTSTATUS_CNF, TAPI_EVENT_NETTEXT_GET_PARAM_CNF, TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF, TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF,

  TAPI_EVENT_NETTEXT_SET_REQUEST_CNF, TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF, TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF, TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF, TAPI_EVENT_NETTEXT_INCOM_IND, TAPI_EVENT_NETTEXT_CB_INCOM_IND, TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND, TAPI_EVENT_NETWORK_SELECT_CNF,

TAPI_EVENT_NETWORK_SEARCH_CNF, TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF, TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF, TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF,

TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF, TAPI_EVENT_NETWORK_SETNWORDER_CNF, TAPI_EVENT_NETWORK_SETNWMODE_CNF, TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF,

TAPI_EVENT_NETWORK_SETNWBAND_CNF, TAPI_EVENT_NETWORK_SETSERVICEDOMAIN_CNF, TAPI_EVENT_NETWORK_CONFIG_CNF, TAPI_EVENT_NETWORK_CHANGE_IND,

TAPI_EVENT_NETWORK_SEARCH_IND, TAPI_EVENT_NETWORK_RADIO_INFO_IND, TAPI_EVENT_SS_BARRING_CNF, TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF,

TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF, TAPI_EVENT_SS_FORWARD_CNF, TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF, TAPI_EVENT_SS_WAITING_CNF,

TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF, TAPI_EVENT_SS_USSD_CNF, TAPI_EVENT_SS_USSDCNF_RSP, TAPI_EVENT_SS_USSDCNF_NOTIFY,

TAPI_EVENT_SS_USSD_IND_RSP, TAPI_EVENT_SS_VERIFY_PWD_CNF, TAPI_EVENT_SS_QUERYCAPABILITY_INFO_CNF, TAPI_EVENT_SS_USSD_IND,

TAPI_EVENT_SS_USSD_IND_NOTIFY, TAPI_EVENT_SS_GET_PASSWORD_IND, TAPI_EVENT_SIM_FILE_INFO_CNF, TAPI_EVENT_SIM_READ_FILE_CNF,

TAPI_EVENT_SIM_UPDATE_FILE_CNF, TAPI_EVENT_SIM_VERIFY_PIN_CNF, TAPI_EVENT_SIM_CHANGE_PIN_CNF, TAPI_EVENT_SIM_DISABLE_PIN_CNF,

TAPI_EVENT_SIM_ENABLE_PIN_CNF, TAPI_EVENT_SIM_UNBLOCK_PIN_CNF, TAPI_EVENT_SIM_DISABLE_FDN_CNF, TAPI_EVENT_SIM_ENABLE_FDN_CNF,

TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF, TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF, TAPI_EVENT_SIM_PB_ACCESS_READ_CNF, TAPI_EVENT_SIM_PB_UPDATE_CNF,

TAPI_EVENT_SIM_PB_DELETE_CNF, TAPI_EVENT_SIM_STATUS_IND, TAPI_EVENT_SIM_FDN_STATUS_IND, TAPI_EVENT_ERR_RESERVED1_IND,

TAPI_EVENT_ERR_RESERVED2_IND, TAPI_EVENT_ERR_RESERVED3_IND, TAPI_EVENT_ERR_RESERVED4_IND, TAPI_EVENT_ERR_RESERVED5_IND,

TAPI_EVENT_TUNNEL_INFO_CNF, TAPI_EVENT_TUNNEL_INFO_IND }

Functions

- TapiResult_t TelTapiRegisterClass (TelTapiEventClass_t EventClass, unsigned int ∗pSubscriptionId, TelAppCallback AppCb)

  This function is exposed to the TAPI clients to register the callback which needs to be invoked for a particular class of Telephony confirmations/indications. If Application tries to call this TelTapiRegisterClass API for the same event class multiple times (with same callback or different callback), multiple callback registration will happen. SubscriptionID is returned back to the application, if the registration procedure is successful. This subscription id needs to be used for de-registering the callback.

- TapiResult_t TelTapiRegisterEvent (TelTapiEventType_t EventType, unsigned int ∗pSubscriptionId, TelAppCallback AppCb)

  This function is used to register the callback function for a specific TAPI Event. If Application tries to register a callback for a specific event multiple times, then multiple callbacks will be registered. When an event indication is received, then registered callbacks for the specific event will be invoked multiple times.

- TapiResult_t TelTapiDeRegister (unsigned int SubscriptionId)

  This function will de-register the application callback registered for the specific event or event class based on the subscription id. This identifier is the same subscription identifier which is returned back to the application during TAPI registration procedure. If de-register API is called during any service requests, Application registered callback will be cleared based on the subscription id. No application callback shall be invoked in case any confirmations/indications are received from TS.

- TapiResult_t TelTapiTunnelReq (const TelTapiTunnelInfo_t ∗pTunnelInfo, int ∗pRequestId)

  This API allows Application to send any proprietary information to OEM.

- TapiResult_t TelTapiGetVersionInfo (const TelTapiVersionInfo_t ∗pVersionInfo)

  This function returns the version information of Telephony Server.

7.8 TelCall.h File Reference

7.8.1 Detailed Description

This file serves as a "C" header file defines structures for Tapi Call Services.

It contains a sample set of constants, enums, structs that would be required by applications.

==

Data Structures

- struct TelCallCugInfo_t
- struct TelCallingNameInfo_t
- struct TelCallUusInfo_t
- struct TelCallSetupParams_t
- struct TelCallStatus_t
- struct TelCallUserDataInfo_t
- struct TelCallEndInfo_t
- struct TelCallRedirectionInfo_t
- struct TelCallIncomingCallInfo_t

Defines

- #define TAPI_CALLING_NAME_SIZE_MAX 80
- #define TAPI_CALL_UUS_DATA_LEN_MAX 131

Enumerations

- enum TelCallCause_t {

  TAPI_CAUSE_SUCCESS, TAPI_CAUSE_ACCESSDISCARD, TAPI_CAUSE_BARR_BYOPERATOR, TAPI_CAUSE_BEARERMODE_NOTIMPL,

  TAPI_CAUSE_BEARERMODE_RESTRICTED, TAPI_CAUSE_BEARERMODE_UNAUTH, TAPI_CAUSE_BEARERMODE_UNAVAIL, TAPI_CAUSE_BUSY,

  TAPI_CAUSE_CALLMETER_EXPIRED, TAPI_CAUSE_CALLNO_ERROR, TAPI_CAUSE_CKTUNAVAIL, TAPI_CAUSE_CONGESTION,

  TAPI_CAUSE_NO_CIRCUIT_AVAIL, TAPI_CAUSE_DESTIN_INCOMPAT, TAPI_CAUSE_DTMF_NOSPEECH, TAPI_CAUSE_DTMF_REJECTED,

  TAPI_CAUSE_FACILITY_NOTIMPL, TAPI_CAUSE_FACILITY_NOTSUBSCRIBED, TAPI_CAUSE_INCOMINGCUGCALLS_BARRED, TAPI_CAUSE_INVALNUM,

  TAPI_CAUSE_MPTY_ERROR, TAPI_CAUSE_NOANSWER, TAPI_CAUSE_NONCUGMEMBER, TAPI_CAUSE_NUMBERCHANGED,

  TAPI_CAUSE_NUMBER_ERROR, TAPI_CAUSE_NWFAILURE, TAPI_CAUSE_OPERATIONUNAVAIL, TAPI_CAUSE_OUTOFORDER,

  TAPI_CAUSE_PROTOCOL_ERROR, TAPI_CAUSE_QOSUNAVAIL, TAPI_CAUSE_REJECT, TAPI_CAUSE_REJ_FAIL,

  TAPI_CAUSE_REJ_SRVC_NOT_AVL, TAPI_CAUSE_REMOTE_CKTUNAVAIL, TAPI_CAUSE_RESOURCEUNAVAIL, TAPI_CAUSE_SERVICEID_ERROR, TAPI_CAUSE_SERVICE_NOTIMPL, TAPI_CAUSE_SERVICE_UNAVAIL, TAPI_CAUSE_MODIFY_SAME_BEARER, TAPI_CAUSE_MODIFY_NOT_PERMITTED,

TAPI_CAUSE_HOLD_FAIL, TAPI_CAUSE_TEMPFAILURE, TAPI_CAUSE_WRONGNUM, TAPI_CAUSE_NORMAL_UNSPECIFIED,

TAPI_CAUSE_SSERRINVALIDTIVALUE, TAPI_CAUSE_SSERRINVALIDTI, TAPI_CAUSE_SSINCOMPATIBLESTATE, TAPI_CAUSE_SSERRSYNTAXERROR,

TAPI_CAUSE_SSPROTOCOLERROR, TAPI_CAUSE_SSNEGATIVEPASSWORDCHECK, TAPI_CAUSE_SSSENTASPHASE1, TAPI_CAUSE_SSERROR,

TAPI_CAUSE_SS_USSD_BUSY, TAPI_CAUSE_IDLE, TAPI_CAUSE_NETWORK_SPECIFIC, TAPI_CAUSE_FADE,

TAPI_CAUSE_UNKNOWN, TAPI_CAUSE_INCOM_CALL, TAPI_CAUSE_ALERT_STOP, TAPI_CAUSE_INTERCEPT,

TAPI_CAUSE_REORDER, TAPI_CAUSE_CLIENT_END, TAPI_CAUSE_ACCESS_CLASS_BLOCKED, TAPI_CAUSE_MEMCAPEXCEEDED,

TAPI_CAUSE_TYPENOTSUPPORTED, TAPI_CAUSE_REPLACENOTSUPPORTED, TAPI_CAUSE_PROTOCOLID_ERROR, TAPI_CAUSE_CLASSNOTSUPPORTED,

TAPI_CAUSE_DATACODING_ERROR, TAPI_CAUSE_INVAL_MSG, TAPI_CAUSE_CALL_BARRED, TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_SS,

TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_CALL, TAPI_CAUSE_SIM_ERROR, TAPI_CAUSE_SIM_CALL_CONTROL_MODIFIED, TAPI_CAUSE_RANDOM_ACCESS_FAILURE,

TAPI_CAUSE_LOWER_LAYER_FAILURE, TAPI_CAUSE_ACCESS_REJECTED, TAPI_CAUSE_REEST_NOT_ALLOWED, TAPI_CAUSE_EMERGENCY_CALL_NOT_ALLOWED,

TAPI_CAUSE_ACCESS_NOT_ALLOWED, TAPI_CAUSE_RADIO_LINK_FAILURE, TAPI_CAUSE_INVALID_IDENTITY, TAPI_CAUSE_UNKNOWN_IDENTITY,

TAPI_CAUSE_UNACCEPTABLE_IDENTITY, TAPI_CAUSE_NO_SERVICE_HERE, TAPI_CAUSE_SERVICE_NOT_ALLOWED, TAPI_CAUSE_SERVICE_NOT_AVAIL,

TAPI_CAUSE_SERVICE_NOT_WORKING, TAPI_CAUSE_CANNOT_IDENTIFY_CALL, TAPI_CAUSE_DEACTIVATION, TAPI_CAUSE_FATAL_ERROR }

- enum TelCallAnswerType_t { TAPI_CALL_ANSWER_ACCEPT = 0, TAPI_CALL_ANSWER_REJECT = 1 }
- enum TelCallType_t { TAPI_CALL_TYPE_VOICE, TAPI_CALL_TYPE_DATA, TAPI_CALL_TYPE_E911 }
- enum TelCallAlertingPattern_t {

  TAPI_CALL_ALERTING_PATTERN_1 = 0x00, TAPI_CALL_ALERTING_PATTERN_2 = 0x01, TAPI_CALL_ALERTING_PATTERN_3 = 0x02, TAPI_CALL_ALERTING_PATTERN_4 = 0x03,

  TAPI_CALL_ALERTING_PATTERN_5 = 0x04, TAPI_CALL_ALERTING_PATTERN_6 = 0x05, TAPI_CALL_ALERTING_PATTERN_7 = 0x06, TAPI_CALL_ALERTING_PATTERN_8 = 0x07,

  TAPI_CALL_NO_ALERTING_PATTERN = 0x08 }
- enum TelCallActiveState_t { TAPI_CALL_CONNECT_IDLE, TAPI_CALL_CONNECT_ACTIVE, TAPI_CALL_CONNECT_HELD }
- enum TelCallIdentityMode_t { TAPI_CALL_IDENTITY_DEFAULT, TAPI_CALL_IDENTITY_SHOW, TAPI_CALL_IDENTITY_HIDE }
- enum TelCallCugOptions_t { TAPI_CALL_CUG_NO_INFO = 0x00, TAPI_CALL_CUG_SUPRESS_OA, TAPI_CALL_CUG_SUPRESS_PRF_CUG, TAPI_CALL_CUG_SUPRESS_OA_AND_CUG }
- enum TelCallUusType_t { TAPI_CALL_UUS_NONE, TAPI_CALL_UUS_1 }

- enum TelCallUusProtocolType_t {
  TAPI_CALL_UUS_PROTO_SPECIFIC = 0, TAPI_CALL_UUS_PROTO_OSI_HIGHER_LAYER = 1, TAPI_CALL_UUS_PROTO_IA5_CHARS = 4, TAPI_CALL_UUS_PROTO_V120 = 7,
  TAPI_CALL_UUS_PROTO_Q931 = 8 }
- enum TelCallNameMode_t { TAPI_CALL_NAME_AVAIL = 0, TAPI_CALL_NAME_RESTRICTED = 1, TAPI_CALL_NAME_UNAVAIL = 2, TAPI_CALL_NAME_AVAIL_RESTRICTED = 3 }

7.9 TelNetText.h File Reference

7.9.1 Detailed Description

This file serves as a "C" header file defines structures for Tapi Network Text Services.

It contains a sample set of constants, enums,structs that would be required by applications.

==

```
#include "ITapiSim.h"
#include "tapi_utility.h"
#include "TapiCommon.h"
```

Data Structures

- struct TelSmsCbConfig_t
- struct TelSmsAddressInfo_t
- struct TelSmsParams_t
- struct TelSmsDatapackageInfo_t
- struct TelSmsData_t
- struct TelSmsStoredMsgCountInfo_t
- struct TelSmsCbMsg_t

Defines

- #define TAPI_NETTEXT_CB_PAGE_SIZE_MAX 93
- #define TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX 90
- #define TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX 60
- #define TAPI_NETTEXT_SMDATA_SIZE_MAX 160
- #define **TAPI_NETTEXT_MAX_GSM_SMS_SERVICE_CENTER_ADDR** 12
- #define **TAPI_NETTEXT_MAX_INDEX** 255
- #define TAPI_SIM_SMSP_ADDRESS_LEN 20
- #define TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX 32

Enumerations

- enum TelSmsMsgStatus_t {

  TAPI_NETTEXT_STATUS_READ, TAPI_NETTEXT_STATUS_UNREAD, TAPI_NETTEXT_STATUS_SENT, TAPI_NETTEXT_STATUS_UNSENT,

  TAPI_NETTEXT_STATUS_DELIVERED, TAPI_NETTEXT_STATUS_DELIVERY_UNCONFIRMED, TAPI_NETTEXT_STATUS_MESSAGE_REPLACED, TAPI_NETTEXT_STATUS_RESERVED }
- enum TelSmsBearerType_t {

  TAPI_NETTEXT_BEARER_PS_ONLY = 0x01, TAPI_NETTEXT_BEARER_CS_ONLY, TAPI_NETTEXT_BEARER_PS_PREFERRED, TAPI_NETTEXT_BEARER_CS_PREFERRED,

  TAPI_NETTEXT_NO_PREFERRED_BEARER }
- enum TelSmsCbMsgType_t { TAPI_NETTEXT_CB_MSG_CBS = 1, TAPI_NETTEXT_CB_MSG_SCHEDULE, TAPI_NETTEXT_CB_MSG_CBS41, TAPI_NETTEXT_CB_MSG_INVALID }

- enum TelSmsResponse_t {

  TAPI_NETTEXT_SENDSMS_SUCCESS = 0x00, TAPI_NETTEXT_ROUTING_NOT_AVAILABLE, TAPI_NETTEXT_INVALID_PARAMETER, TAPI_NETTEXT_DEVICE_FAILURE,

  TAPI_NETTEXT_SERVICE_RESERVED, TAPI_NETTEXT_INVALID_LOCATION, TAPI_NETTEXT_NO_SIM, TAPI_NETTEXT_SIM_NOT_READY,

  TAPI_NETTEXT_NO_NETWORK_RESP, TAPI_NETTEXT_DEST_ADDRESS_FDN_RESTRICTED, TAPI_NETTEXT_SCADDRESS_FDN_RESTRICTED, TAPI_NETTEXT_RESEND_ALREADY_DONE,

  TAPI_NETTEXT_SCADDRESS_NOT_AVAILABLE, TAPI_NETTEXT_UNASSIGNED_NUMBER, TAPI_NETTEXT_OPERATOR_DETERMINED_BARRING, TAPI_NETTEXT_CALL_BARRED,

  TAPI_NETTEXT_MESSAGE_TRANSFER_REJECTED, TAPI_NETTEXT_MEMORY_CAPACITY_EXCEEDED, TAPI_NETTEXT_DESTINAITION_OUTOFSERVICE, TAPI_NETTEXT_UNSPECIFIED_SUBSCRIBER,

  TAPI_NETTEXT_FACILITY_REJECTED, TAPI_NETTEXT_UNKNOWN_SUBSCRIBER, TAPI_NETTEXT_NETWORK_OUTOFORDER, TAPI_NETTEXT_TEMPORARY_FAILURE,

  TAPI_NETTEXT_CONGESTION, TAPI_NETTEXT_RESOURCES_UNAVAILABLE, TAPI_NETTEXT_FACILITY_NOT_SUBSCRIBED, TAPI_NETTEXT_FACILITY_NOT_IMPLEMENTED,

  TAPI_NETTEXT_INVALID_REF_VALUE, TAPI_NETTEXT_INVALID_MSG, TAPI_NETTEXT_INVALID_MANDATORY_INFO, TAPI_NETTEXT_MESSAGE_TYPE_NOT_IMPLEMENTED,

  TAPI_NETTEXT_MESSAGE_NOT_COMPAT_PROTOCOL, TAPI_NETTEXT_IE_NOT_IMPLEMENTED, TAPI_NETTEXT_PROTOCOL_ERROR, TAPI_NETTEXT_INTERWORKING }
- enum TelSmsCause_t {

  TAPI_NETTEXT_SUCCESS, TAPI_NETTEXT_INVALID_PARAMETER_FORMAT, TAPI_NETTEXT_PHONE_FAILURE, TAPI_NETTEXT_SIM_BUSY,

  TAPI_NETTEXT_SIM_FAILURE, TAPI_NETTEXT_UNKNOWN, TAPI_NETTEXT_MEMORY_FAILURE, TAPI_NETTEXT_OPERATION_NOT_SUPPORTED }

7.10 TelNetwork.h File Reference

7.10.1 Detailed Description

This file serves as a "C" header file defines structures for Tapi Network Services.

It contains a sample set of constants, enums,structs that would be required by applications.

==

Data Structures

- struct TelNetworkTimeAndZone_t
- struct TelNetworkSystemId_t
- struct TelNetworkConfigSettings_t
- struct TelNetworkInfo_t
- struct TelNetworkList_t
- struct TelNetworkRadioInfo_t

Defines

- #define TAPI_NETWORK_ORDER_MAX 5
- #define TAPI_NETWORK_LIST_MAX 15
- #define TAPI_NETWORK_NAME_LEN_MAX 40
- #define TAPI_NETWORK_SPN_LEN_MAX 20

Enumerations

- enum TelNetworkSelectionMode_t { TAPI_NETWORK_SELECTIONMODE_AUTOMATIC, TAPI_NETWORK_SELECTIONMODE_MANUAL }
- enum TelNetworkSystemIdFormat_t { TAPI_NETWORK_SYSID_FORMAT_SID, TAPI_NETWORK_SYSID_FORMAT_PLMN }
- enum TelNetworkServiceDomain_t { TAPI_NETWORK_SERVICE_DOMAIN_COMBINED, TAPI_NETWORK_SERVICE_DOMAIN_PS, TAPI_NETWORK_SERVICE_DOMAIN_CS, TAPI_NETWORK_SERVICE_DOMAIN_AUTOMATIC }
- enum TelNetworkMode_t {

  TAPI_NETWORK_MODE_EMERGENCY, TAPI_NETWORK_MODE_AUTOMATIC, TAPI_NETWORK_MODE_CDMA, TAPI_NETWORK_MODE_GSM,

  TAPI_NETWORK_MODE_WCDMA, TAPI_NETWORK_MODE_WLAN }
- enum TelNetworkServiceLevel_t { TAPI_NETWORK_SERVICE_LEVEL_NO, TAPI_NETWORK_SERVICE_LEVEL_EMERGENCY, TAPI_NETWORK_SERVICE_LEVEL_FULL, TAPI_NETWORK_SERVICE_LEVEL_SEARCH }
- enum TelNetworkServiceType_t {

  TAPI_NETWORK_SERVICE_TYPE_NO_SVE, TAPI_NETWORK_SERVICE_TYPE_LIMITED, TAPI_NETWORK_SERVICE_TYPE_CS, TAPI_NETWORK_SERVICE_TYPE_PS,

  TAPI_NETWORK_SERVICE_TYPE_COMBINED }

- enum TelNetworkSystemType_t {

  TAPI_NETWORK_SYSTEM_NO_SRV, TAPI_NETWORK_SYSTEM_CDMA, TAPI_NETWORK_SYSTEM_TDMA, TAPI_NETWORK_SYSTEM_GSM,

  TAPI_NETWORK_SYSTEM_GPRS, TAPI_NETWORK_SYSTEM_EGPRS, TAPI_NETWORK_SYSTEM_PCS1900, TAPI_NETWORK_SYSTEM_UMTS,

  TAPI_NETWORK_SYSTEM_GSM_AND_UMTS }
- enum TelNetworkBandPreferred_t { TAPI_NETWORK_BAND_MODE_PREFERRED = 0x01, TAPI_NETWORK_BAND_MODE_ONLY = 0x02 }
- enum TelNetworkBand_t {

  TAPI_NETWORK_BAND_TYPE_ANY = 0x00, TAPI_NETWORK_BAND_TYPE_GSM850, TAPI_NETWORK_BAND_TYPE_GSM_900_1800, TAPI_NETWORK_BAND_TYPE_GSM1900,

  TAPI_NETWORK_BAND_TYPE_GSM, TAPI_NETWORK_BAND_TYPE_WCDMA, TAPI_NETWORK_BAND_TYPE_WCDMA850, TAPI_NETWORK_BAND_TYPE_WCDMA1900,

  TAPI_NETWORK_BAND_TYPE_WCDMA2100 }
- enum TelNetworkOperationCause_t {

  TAPI_NETWORK_NO_ERR = 0x00, TAPI_NETWORK_OPERATION_FAILED, TAPI_NETWORK_PHONE_IN_USE_ERR, TAPI_NETWORK_OFFLINE_ERR,

  TAPI_NETWORK_CONFIG_SETTINGS_FAILURE_ERR, TAPI_NETWORK_INTERNAL_FAILURE_ERR }
- enum TelNetworkDayLightSavingAdj_t { TAPI_NETWORK_DLS_ADJ_NO_ADJUSTMENT, TAPI_NETWORK_DLS_ADJ_PLUS_ONE_HOUR, TAPI_NETWORK_DLS_PLUS_TWO_HOUR }

7.11 TelSim.h File Reference

7.11.1 Detailed Description

This file serves as a "C" header file defines structures for Tapi SIM Services.

It contains a sample set of constants, enums,structs that would be required by applications.

==

`#include "limo_defines.h"`

`#include "TapiCommon.h"`

Data Structures

- struct TelSimFileID_t
- struct TelSimPinData_t
- struct TelSimReadFileRequestData_t
- struct TelSimIccIdInfo_t
- struct TelSimImsiInfo_t
- struct TelSimServiceTable_t
- struct TelSimSstInfo_t
- struct TelSimPinInfo_t
- struct TelSimPb2gData_t
- struct TelSimPbRecordData_t
- struct TelSimEccInfo_t
- struct TelUsimEccInfo_t
- struct TelUSimServiceTable_t
- struct TelSimUstInfo_t
- struct TelSimPbStorageInfo_t
- struct TelSimPbEntryInfo_t
- struct TelSimPbReadResp_t
- struct TelSimPbUpdateResp_t
- struct TelSimLanguageInfo_t
- struct TelSimFileAccessGeneralResp_t
- struct TelSimGroupIdentifierInfo_t
- struct TelSimEccData_t
- struct TelSimStatusData_t
- struct TelSimCallbackEvent_t
- struct TelSimEvent_t
- struct TelSimApnData_t
- struct TelSimDeviceContextData_t
- struct TelSimDeviceContext_t
- struct TelSimFileInfo_t
- struct TelSimReadFileData_t
- struct TelSimUpdateFileRequestData_t

Defines

- #define TAPI_SIM_PIN_LENGTH 8
- #define TAPI_SIM_ICCIDLENGTH 10
- #define TAPI_SIM_ICCID_PACKET_LEN 10
- #define TAPI_SIM_ICCID_MAX_LEN 10
- #define TAPI_SIM_LP_MAX_LEN 30
- #define TAPI_SIM_LP_PACKET_MAX_LEN 30
- #define TAPI_SIM_IMSI_LEN 8
- #define TAPI_SIM_IMSI_PACKET_LEN 15
- #define TAPI_SIM_MCC_CODE_LEN 3
- #define TAPI_SIM_MNC_CODE_LEN 2
- #define TAPI_SIM_MSIN_CODE_LEN 10
- #define TAPI_SIM_NUM_TEXT_MAX_LEN 256
- #define TAPI_SIM_SST_MAX_LEN 20
- #define TAPI_SIM_GSM_SST_SERVICE_MAX_COUNT 50
- #define TAPI_SIM_ECC_CODE_LEN 3
- #define TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX 50
- #define TAPI_TAPI_SIM_PBOOK_ENTRY_COUNT_MAX 20
- #define TAPI_SIM_LI_MAX_LEN 30
- #define TAPI_SIM_LI_PACKET_MAX_LEN 30
- #define TAPI_SIM_ACL_MAX_LEN 50
- #define TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX 10
- #define TAPI_SIM_ECC_MAX_LEN 15
- #define TAPI_SIM_UECC_RECORD_COUNT_MAX 5
- #define TAPI_SIM_UECC_RECORD_LEN_MAX 50

Enumerations

- enum TelSimCardType_t { TAPI_SIM_CARD_TYPE_UNKNOWN, TAPI_SIM_CARD_TYPE_GSM, TAPI_SIM_CARD_TYPE_USIM }
- enum TelSimFileName_t {

  TAPI_SIM_EFILE_DIR = 0x2F00, TAPI_SIM_EFILE_ICCID = 0x2FE2, TAPI_SIM_EFILE_ADN = 0x6F3A, TAPI_SIM_EFILE_FDN = 0x6F3B,

  TAPI_SIM_EFILE_SMS = 0x6F3C, TAPI_SIM_EFILE_CCP = 0x6F3D, TAPI_SIM_EFILE_IMSI = 0x6F07, TAPI_SIM_EFILE_SST = 0x6F38,

  TAPI_SIM_EFILE_EST = 0x6F56, TAPI_SIM_EFILE_ACM = 0x6F39, TAPI_SIM_EFILE_MBDN = 0x6FC7, TAPI_SIM_EFILE_GID1 = 0x6F3E,

  TAPI_SIM_EFILE_GID2 = 0x6F3F, TAPI_SIM_EFILE_MSISDN = 0x6F40, TAPI_SIM_EFILE_SMSP = 0x6F42, TAPI_SIM_EFILE_SMSS = 0x6F43,

  TAPI_SIM_EFILE_EXT1 = 0x6F4A, TAPI_SIM_EFILE_EXT2 = 0x6F4B, TAPI_SIM_EFILE_LP = 0x2F05, TAPI_SIM_EFILE_ACM_MAX = 0x6F37,

  TAPI_SIM_EFILE_PUCT = 0x6F41, TAPI_SIM_EFILE_CBMI = 0x6F45, TAPI_SIM_EFILE_PHASE = 0x6FAE, TAPI_SIM_EFILE_SDN = 0x6F49,

  TAPI_SIM_EFILE_EXT3 = 0x6F4C, TAPI_SIM_EFILE_BDN = 0x6F4D, TAPI_SIM_EFILE_EXT4 = 0x6F4E, TAPI_SIM_EFILE_ECC = 0x6FB7,

  TAPI_SIM_EFILE_SPN = 0x6F46, TAPI_SIM_EFILE_PNN = 0x6FC5, TAPI_SIM_EFILE_IMG = 0x4F20, TAPI_SIM_EFILE_USIM_LI = 0x6F05, TAPI_SIM_EFILE_USIM_IMSI = 0x6A07, TAPI_SIM_EFILE_USIM_ACM_MAX = 0x6A37, TAPI_SIM_EFILE_USIM_UST = 0x6A38, TAPI_SIM_EFILE_USIM_ACM = 0x6A39,

TAPI_SIM_EFILE_USIM_SPN = 0x6A46, TAPI_SIM_EFILE_USIM_PUCT = 0x6A41, TAPI_SIM_EFILE_USIM_CBMI = 0x6A45, TAPI_SIM_EFILE_USIM_ECC = 0x6AB7,

TAPI_SIM_EFILE_USIM_FDN = 0x6A3B, TAPI_SIM_EFILE_USIM_SMS = 0x6A3C, TAPI_SIM_EFILE_USIM_MSISDN = 0x6A40, TAPI_SIM_EFILE_USIM_SMSP = 0x6A42,

TAPI_SIM_EFILE_USIM_SMSS = 0x6A43, TAPI_SIM_EFILE_USIM_SDN = 0x6A49, TAPI_SIM_EFILE_USIM_EXT1 = 0X6A4A, TAPI_SIM_EFILE_USIM_EXT2 = 0x6A4B,

TAPI_SIM_EFILE_USIM_EXT3 = 0x6A4C, TAPI_SIM_EFILE_USIM_EXT4 = 0x6A55, TAPI_SIM_EFILE_USIM_EXT5 = 0x6A4E, **TAPI_SIM_EFILE_USIM_EXT6** = 0x6FC8,

TAPI_SIM_EFILE_USIM_CCP2 = 0x6A4F, TAPI_SIM_EFILE_USIM_HIDDEN_KEY = 0x6FC3, TAPI_SIM_EFILE_USIM_BDN = 0x6A4D, TAPI_SIM_EFILE_USIM_MBDN = 0x6FC7,

TAPI_SIM_EFILE_USIM_MBI = 0x6FC9, TAPI_SIM_EFILE_USIM_PHONEBOOK_PBR = 0x4FF0, TAPI_SIM_EFILE_USIM_PHONEBOOK_CCP1 = 0x4FFE, TAPI_SIM_EFILE_USIM_PHONEBOOK_UID = 0x4FF8,

TAPI_SIM_EFILE_USIM_PHONEBOOK_PSC = 0x4FFF, TAPI_SIM_EFILE_USIM_PHONEBOOK_CC = 0x4FEF, TAPI_SIM_EFILE_USIM_PHONEBOOK_PUID = 0x4FEE, TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN = 0x4FF1,

TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP = 0x4FF2, TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC = 0x4FF3, TAPI_SIM_EFILE_USIM_PHONEBOOK_EXT1 = 0x4FFB, TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP = 0x4FF7,

TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS = 0x4FFC, TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS1 = 0x4FFD, TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS = 0x4F0E, TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS1 = 0x4F0F,

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR = 0x4FF4, TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE = 0x4FF9, TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL = 0x4FFA, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA = 0x4F13,

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB = 0x4F14, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC = 0x4F15, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR1 = 0x4F16, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA1 = 0x4F17,

TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB1 = 0x4F18, TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC1 = 0x4F21, TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN1 = 0x4F1A, TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC1 = 0x4F1B,

TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP1 = 0x4F1C, TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE1 = 0x4F1D, TAPI_SIM_EFILE_USIM_PHONEBOOK_UID1 = 0x4F1E, TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL1 = 0x4F1F,

TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP1 = 0x4FF5, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR = 0x4F30, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR1 = 0x4F29, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR2 = 0x4F28,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CCP1 = 0x4F3D, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID = 0x4F00, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PSC = 0x4F22, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CC = 0x4F23,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PUID = 0x4F24, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN = 0x4F3A, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP = 0x4F25, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC = 0x4F09,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EXT1 = 0x4F4A, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP = 0x4F26, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_AAS = 0x4F4B, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GAS = 0x4F4C, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR = 0x4F11, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE = 0x4F19, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL = 0x4F50, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA = 0x4F31,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB = 0x4F32, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRC = 0x4F40, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR1 = 0x4F33, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA1 = 0x4F34,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB1 = 0x4F35, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN1 = 0x4F36, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC1 = 0x4F37, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP1 = 0x4F38,

TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE1 = 0x4F39, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID1 = 0x4F3B, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL1 = 0x4F3C, TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP1 = 0x4F3E,

TAPI_SIM_EFILE_MSG_WAITING_INDICATION_STATUS = 0x6FCA, TAPI_SIM_EFILE_INVALID = 0xFFFF, TAPI_SIM_EFILE_OTHERS }

- enum TelSimFileType_t {

  TAPI_SIM_FTYPE_DEDICATED = 0x00, TAPI_SIM_FTYPE_TRANSPARENT = 0x01, TAPI_SIM_FTYPE_LINEAR_FIXED = 0x02, TAPI_SIM_FTYPE_CYCLIC = 0x04,

  TAPI_SIM_FTYPE_INVALID_TYPE = 0xFF }

- enum TelSimRecordMode_t {

  TAPI_SIM_RECORD_NONE = 0x00, TAPI_SIM_RECORD_ABSOLUTE = 0x01, TAPI_SIM_RECORD_CURRENT = 0x02, TAPI_SIM_RECORD_NEXT = 0x03,

  TAPI_SIM_RECORD_PREVIOUS = 0x04, TAPI_SIM_RECORD_TO_THE_LAST = 0x05, TAPI_SIM_RECORD_FROM_THE_FIRST = 0x06 }

- enum TelSimPinStatus_t {

  TAPI_SIM_PIN_STATUS_DISABLED = 0x00, TAPI_SIM_PIN_STATUS_ENABLED = 0x01, TAPI_SIM_PIN_STATUS_BLOCKED = 0x02, TAPI_SIM_PIN_STATUS_PUK_BLOCKED = 0x03,

  TAPI_SIM_PIN_STATUS_UNKNOWN = 0xFF }

- enum TelSimPinOperationResult_t {

  TAPI_SIM_PIN_OPERATION_SUCCESS, TAPI_SIM_BUSY_STATUS, TAPI_SIM_CARD_ERROR_STATUS, TAPI_SIM_INCOMPATIBLE_PIN_OPERATION_STATUS,

  TAPI_SIM_PIN_INCORRECT_PASSWORD_STATUS, TAPI_SIM_PIN2_REQD_STATUS, TAPI_SIM_PIN2_INCORRECT_PASSWORD_STATUS, TAPI_SIM_PUK_INCORRECT_PASSWORD_STATUS,

  TAPI_SIM_PUK_REQD_STATUS, TAPI_SIM_PUK2_REQD_STATUS }

- enum TelSimAccessResult_t {

  TAPI_SIM_ACCESS_SUCCESS, TAPI_SIM_ACCESS_CARD_ERROR, TAPI_SIM_ACCESS_FILE_NOT_FOUND, TAPI_SIM_ACCESS_ACCESS_CONDITION_NOT_SATISFIED,

  TAPI_SIM_ACCESS_FAILED }

- enum TelSimPinType_t {

  TAPI_SIM_PTYPE_PIN1 = 0x00, TAPI_SIM_PTYPE_PIN2 = 0x01, TAPI_SIM_PTYPE_PUK1 = 0x02, TAPI_SIM_PTYPE_PUK2 = 0x03,

  TAPI_SIM_PTYPE_UPIN = 0x04, TAPI_SIM_PTYPE_ADM = 0x05 }

- enum TelSimTypeOfNum_t {

  TAPI_SIM_TON_UNKNOWN = 0, TAPI_SIM_TON_INTERNATIONAL = 1, TAPI_SIM_TON_NATIONAL = 2, TAPI_SIM_TON_NETWORK_SPECIFIC = 3,

  TAPI_SIM_TON_DEDICATED_ACCESS = 4, TAPI_SIM_TON_ALPHA_NUMERIC = 5, TAPI_SIM_TON_ABBREVIATED_NUMBER = 6, TAPI_SIM_TON_RESERVED_FOR_EXT = 7 }

- enum TelSimTextEncrypt_t { TAPI_SIM_TEXT_ENC_ASCII, TAPI_SIM_TEXT_ENC_GSM7BIT, TAPI_SIM_TEXT_ENC_UCS2, TAPI_SIM_TEXT_ENC_HEX }
- enum TelSimPbFileType_t {

  TAPI_SIM_PB_FDN, TAPI_SIM_PB_LDN, TAPI_SIM_PB_MSISDN, TAPI_SIM_PB_ADN,

  TAPI_SIM_PB_SDN }
- enum TelSimPbAccessMode_t { TAPI_SIM_PB_ACCESS_MODE_ADD = 0x01, TAPI_SIM_PB_ACCESS_MODE_DEL = 0x02, TAPI_SIM_PB_ACCESS_MODE_EDIT = 0x03 }
- enum TelSimNumberingPlanIdentity_t {

  TAPI_SIM_NPI_UNKNOWN = 0, TAPI_SIM_NPI_ISDN_TEL = 1, TAPI_SIM_NPI_DATA_NUMBERING_PLAN = 3, TAPI_SIM_NPI_TELEX = 4,

  TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN = 5, TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN2 = 6, TAPI_SIM_NPI_NATIONAL = 8, TAPI_SIM_NPI_PRIVATE = 9,

  TAPI_SIM_NPI_ERMES_NUMBERING_PLAN = 10, TAPI_SIM_NPI_RESERVED_FOR_EXT = 0xF }
- enum TelSimEccEmergencyServiceInfo_t {

  TAPI_SIM_ECC_ESC_POLICE = 0x01, TAPI_SIM_ECC_ESC_AMBULANCE = 0x02, TAPI_SIM_ECC_ESC_FIREBRIGADE = 0x03, TAPI_SIM_ECC_ESC_MARAINEGUARD = 0x04,

  TAPI_SIM_ECC_ESC_MOUTAINRESCUE = 0x05, TAPI_SIM_ECC_ESC_SPARE = 0x00 }
- enum TelSimPbAccessResult_t {

  TAPI_SIM_PB_SUCCESS, TAPI_SIM_PB_FAIL, TAPI_SIM_PB_INVALID_INDEX, TAPI_SIM_PB_INVALID_NUMBER_LENGTH,

  TAPI_SIM_PB_INVALID_NAME_LENGTH, TAPI_SIM_PB_ACCESS_CONDITION_NOT_SATISFIED }
- enum TelSimLangIndicationCode_t {

  TAPI_SIM_LI_GERMAN = 0, TAPI_SIM_LI_ENGLISH = 0x01, TAPI_SIM_LI_ITALIAN = 0x02, TAPI_SIM_LI_FRENCH = 0x03,

  TAPI_SIM_LI_SPANISH = 0x04, TAPI_SIM_LI_DUTCH = 0x05, TAPI_SIM_LI_SWEDISH = 0x06, TAPI_SIM_LI_DANISH = 0x07,

  TAPI_SIM_LI_PORTUGUESE = 0x08, TAPI_SIM_LI_FINNISH = 0x09, TAPI_SIM_LI_NORWEGIAN = 0x0A, TAPI_SIM_LI_GREEK = 0x0B,

  TAPI_SIM_LI_TURKISH = 0x0C, TAPI_SIM_LI_HUNGARIAN = 0x0D, TAPI_SIM_LI_POLISH = 0x0E, TAPI_SIM_LI_LANG_UNSPECIFIED = 0x0F }
- enum TelSimAccessType_t {

  TAPI_SIM_SELECT, TAPI_SIM_READ_BINARY, TAPI_SIM_READ_RECORD, TAPI_SIM_GET_RESPONSE,

  TAPI_SIM_UPDATE_BINARY, TAPI_SIM_UPDATE_RECORD, TAPI_SIM_STATUS }
- enum TelSimCardStatus_t {

  TAPI_SIM_STATUS_CARD_ERROR = 0x00, TAPI_SIM_STATUS_CARD_NOT_PRESENT = 0x01, TAPI_SIM_STATUS_SIM_INITIALIZING = 0x02, TAPI_SIM_STATUS_SIM_INIT_COMPLETED = 0x03,

  TAPI_SIM_STATUS_SIM_PIN_REQUIRED = 0x04, TAPI_SIM_STATUS_SIM_PUK_REQUIRED = 0x05, TAPI_SIM_STATUS_CARD_BLOCKED = 0x06 }
- enum TelSimAcl_t {

  TAPI_SIM_ACL_ALW = 0x00, TAPI_SIM_ACL_PIN1 = 0x01, TAPI_SIM_ACL_PIN2 = 0x02, TAPI_SIM_ACL_RFU = 0x04,

  TAPI_SIM_ACL_ADM = 0x08, TAPI_SIM_ACL_NEVER = 0xFF }

7.12 TelSs.h File Reference

7.12.1 Detailed Description

This file serves as a "C" header file defines structures for Tapi SS Services.

It contains a sample set of constants, enums,structs that would be required by applications.

==

Data Structures

- struct TelSsForwardInfo_t
- struct TelSsUssdMsgInfo_t
- struct TelSsCallBarringInfo_t
- struct TelSsCliRecord_t
- struct TelSsCallBarringRecord_t
- struct TelSsForwardRecord_t
- struct TelSsCallWaitingRecord_t
- union TelSsRecord_t
- struct TelSsInfo_t
- struct TelSsCapabilityInfo_t

Defines

- #define TAPI_SS_GSM_BARR_PW_LEN_MAX 4
- #define TAPI_CALL_DIALDIGIT_LEN_MAX 82
- #define TAPI_SS_USSD_DATA_SIZE_MAX 201

Enumerations

- enum TelSsForwardWhen_t {

  TAPI_SS_FORWARD_WHEN_UNCONDITIONAL_EV, TAPI_SS_FORWARD_WHEN_BUSY_EV, TAPI_SS_FORWARD_WHEN_NO_ANSWER_EV, TAPI_SS_FORWARD_WHEN_NOT_REACHABLE_EV,

  TAPI_SS_FORWARD_WHEN_ALL_FORWARDING_EV }
- enum TelSsCallBarType_t {

  TAPI_CALL_BARRING_ALL, TAPI_CALL_BARRING_OUTGOING, TAPI_CALL_BARRING_ALL_OUTGOING_CALLS, TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL,

  TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL_EXCEPT, TAPI_CALL_BARRING_INCOMING, TAPI_CALL_BARRING_ALL_INCOMING_CALLS, TAPI_CALL_BARRING_ALL_INCOMING_CALLS_ROAMING,

  TAPI_CALL_BARRING_ALL_INCOMING_CALLS_INSIM }
- enum TelSsTeleService_t {

  TAPI_SS_TS_ALL_TELE_AND_BEARER_SERVICES, TAPI_SS_TS_ALL_TELESERVICES, TAPI_SS_TS_ALL_SPEECH, TAPI_SS_TS_ALL_DATA_TELESERVICES,

  TAPI_SS_TS_FACS, TAPI_SS_TS_ALL_TELESERVICES_EXCEPT_SMS, TAPI_SS_TS_ALL_PLMN_SPEC_TELE, TAPI_SS_TS_PLMN_SPEC_TELE_1, TAPI_SS_TS_PLMN_SPEC_TELE_2, TAPI_SS_TS_PLMN_SPEC_TELE_3, TAPI_SS_TS_PLMN_SPEC_TELE_4, TAPI_SS_TS_PLMN_SPEC_TELE_5,

TAPI_SS_TS_PLMN_SPEC_TELE_6, TAPI_SS_TS_PLMN_SPEC_TELE_7, TAPI_SS_TS_PLMN_SPEC_TELE_8, TAPI_SS_TS_PLMN_SPEC_TELE_9,

TAPI_SS_TS_PLMN_SPEC_TELE_A, TAPI_SS_TS_PLMN_SPEC_TELE_B, TAPI_SS_TS_PLMN_SPEC_TELE_C, TAPI_SS_TS_PLMN_SPEC_TELE_D,

TAPI_SS_TS_PLMN_SPEC_TELE_E, TAPI_SS_TS_PLMN_SPEC_TELE_F, **TAPI_SS_TS_PLMN_SPEC_TELE_**, TAPI_SS_BS_ALL_BEARER_SERVICES,

TAPI_SS_BS_ALL_DATA_CDS, TAPI_SS_BS_ALL_ASYNCHRONOUS, TAPI_SS_BS_ALL_SYNCHRONOUS, TAPI_SS_BS_DATA_CIRCUIT_SYNC,

TAPI_SS_BS_DATA_CIRCUIT_ASYNC, TAPI_SS_BS_DATA_ALL_PDS, TAPI_SS_BS_ALL_DATA_PADCA, TAPI_SS_BS_ALL_PLMN_SPEC_BS,

TAPI_SS_BS_PLMN_SPEC_TELE_1, TAPI_SS_BS_PLMN_SPEC_TELE_2, TAPI_SS_BS_PLMN_SPEC_TELE_3, TAPI_SS_BS_PLMN_SPEC_TELE_4,

TAPI_SS_BS_PLMN_SPEC_TELE_5, TAPI_SS_BS_PLMN_SPEC_TELE_6, TAPI_SS_BS_PLMN_SPEC_TELE_7, TAPI_SS_BS_PLMN_SPEC_TELE_8,

TAPI_SS_BS_PLMN_SPEC_TELE_9, TAPI_SS_BS_PLMN_SPEC_TELE_A, TAPI_SS_BS_PLMN_SPEC_TELE_B, TAPI_SS_BS_PLMN_SPEC_TELE_C,

TAPI_SS_BS_PLMN_SPEC_TELE_D, TAPI_SS_BS_PLMN_SPEC_TELE_E, TAPI_SS_BS_PLMN_SPEC_TELE_F, TAPI_SS_TS_ALL_DATA,

TAPI_SS_TS_DATA_CIRCUIT_SYNC, TAPI_SS_TS_MULTIPLE, TAPI_SS_TS_TELEPHONY, TAPI_SS_TS_DATA_CIRCUIT_ASYNC,

TAPI_SS_TS_DATA_PACKET }

- enum TelSsUssdRespType_t { TAPI_USSD_RES_UNKNOWN_ALPHABET, TAPI_USSD_RES_DISPL_BUSY, TAPI_USSD_RES_OK, TAPI_USSD_RES_NONE }
- enum TelSsUssdIndType_t {

  TAPI_USSD_IND_NOTIFY = 0x00, TAPI_USSD_IND_NO_ACTION_REQUIRE = 0x01, TAPI_USSD_IND_ACTION_REQUIRE = 0x02, TAPI_USSD_IND_TERMINATED_BY_NET = 0x03,

  TAPI_USSD_IND_OTHER_CLIENT = 0x04, TAPI_USSD_IND_NOT_SUPPORT = 0x05, TAPI_USSD_IND_TIME_OUT = 0x06 }
- enum TelSsCallBarOperationMode_t { TAPI_SS_CALL_BAR_ACTIVATE, TAPI_SS_CALL_BAR_DEACTIVATE }
- enum TelSsCallType_t {

  TAPI_CALL_TYPE_VOICE_EV, TAPI_CALL_TYPE_DATA_EV, TAPI_CALL_TYPE_FAX_EV, TAPI_CALL_TYPE_SHORT_MESS_EV,

  TAPI_CALL_TYPE_DATA_CIRCUIT_SYNC_EV }
- enum TelSsForwardMode_t { TAPI_CALL_FORWARD_MODE_DISABLE_EV, TAPI_CALL_FORWARD_MODE_ENABLE_EV, TAPI_CALL_FORWARD_MODE_REGISTRATION_EV, TAPI_CALL_FORWARD_MODE_ERASURE_EV }
- enum TelSsForwardType_t { TAPI_CS_FORWARD_TYPE_VOICE_EV = 0, TAPI_CS_FORWARD_TYPE_DATA_EV, TAPI_CS_FORWARD_TYPE_FAX_EV }
- enum TelSsRestrictIDStatus_t { TAPI_CALL_RESTRICT_ID_DEPEND_NETWORK_EV, TAPI_CALL_RESTRICT_ID_ENABLE_EV, TAPI_CALL_RESTRICT_ID_DISABLE_EV }
- enum TelSsType_t {

  TAPI_SS_TYPE_BARRING, TAPI_SS_TYPE_FORWARDING, TAPI_SS_TYPE_WAITING, TAPI_SS_TYPE_IDENTIFICATION,

  TAPI_SS_TYPE_CNAP, TYPE_SS_TYPE_USSD }

- enum TelSsCause_t {

  TAPI_SS_SUCCESS, TAPI_SS_TIMEREXPIRE, TAPI_SS_UNKNOWNSUBSCRIBER, TAPI_SS_SERVICENOTPROVISIONED,

  TAPI_SS_CALLBARRED, TAPI_SS_ILLEGALSSOPERATION_ERROR, TAPI_SS_NOTAVAILABLE, TAPI_SS_SUBSCRIPTIONVIOLATION,

  TAPI_SS_FACILITYNOTSUPPORTED, TAPI_SS_UNEXPECTEDDATAVALUE, TAPI_SS_PWREGISTRATIONFAILURE, TAPI_SS_NEGATIVEPWCHECK,

  TAPI_SS_NUMBEROFPWATTEMPTSVIOLATION, TAPI_SS_UNKNOWNALPHABET, TAPI_SS_USSDBUSY, TAPI_SS_REJECTEDBYUSER,

  TAPI_SS_REJECTEDBYNETWORK, TAPI_SS_INVALIDDEFLECTEDTONUMBER, TAPI_SS_MAXNOMPTYEXCEEDED, TAPI_SS_RESOURCESNOTAVAILABLE,

  TAPI_SS_UNKNOWNERROR, TAPI_SS_OEM_NOT_SUPPORTED }
- enum TelSsStatus_t {

  TAPI_SS_STATUS_ACTIVE, TAPI_SS_STATUS_REGISTERED, TAPI_SS_STATUS_PROVISIONED, TAPI_SS_STATUS_QUIESCENT,

  TAPI_SS_STATUS_NOTHING }
- enum TelSsLineIdentificationType_t {

  TAPI_CALLING_LINE_IDENTITY_PRESENTATION, TAPI_CALLING_LINE_IDENTITY_RESTRICTION, TAPI_CONNECTED_LINE_IDENTITY_PRESENTATION, TAPI_CONNECTED_LINE_IDENTITY_RESTRICTION,

  TAPI_CALLED_LINE_IDENTITY_PRESENTATION, TAPI_CALLING_NAME_PRESENTATION }
- enum TelSsCliStatus_t {

  TAPI_CLI_STATUS_NOT_PROVISONED, TAPI_CLI_STATUS_PROVISIONED, TAPI_CLI_STATUS_ACTIVATED, TAPI_CLI_STATUS_UNKOWN,

  TAPI_CLI_STATUS_TEMP_RESTRICTED, TAPI_CLI_STATUS_TEMP_ALLOWED }
- enum TelSsBarPwdType_t { TAPI_SS_BAR_GET_PWD, TAPI_SS_BAR_NEW_PWD, TAPI_SS_BAR_VERIFY_NEW_PWD }

7.13 TelUtility.h File Reference

7.13.1 Detailed Description

This file serves as a "C" header file defines structures for Utility Services.

It contains a sample set of constants, enums,structs that would be required by applications.

==

Data Structures

- struct TelTapiVersionInfo_t
- struct TelTapiTunnelInfo_t
- struct TelTapiErrorInfo_t

Enumerations

- enum TapiResult_t {

  TAPI_API_SUCCESS, TAPI_API_INVALID_INPUT, TAPI_API_INVALID_PTR, TAPI_API_INVALID_CALL_STATE,

  TAPI_API_NOT_SUPPORTED, TAPI_API_DEPRICATED, TAPI_API_SYSTEM_OUT_OF_MEM, TAPI_API_SYSTEM_RPC_LINK_DOWN,

  TAPI_API_MODEM_POWERED_OFF, TAPI_API_SERVICE_NOT_READY, TAPI_API_SERVER_FAILURE, TAPI_API_OEM_PLUGIN_FAILURE,

  TAPI_API_INVALID_CALL_HANDLE, TAPI_API_INVALID_DATA_LEN, TAPI_API_SIM_NOT_FOUND, TAPI_API_SIM_BUSY,

  TAPI_API_SIM_INVALID_CARD_TYPE, TAPI_API_SIM_CARD_PERMANENT_ERROR, TAPI_API_SIM_NOT_INITIALIZED, TAPI_API_REQUEST_MAX_IN_PROGRESS,

  TAPI_API_OFFLINE_MODE_ERROR, TAPI_EVENT_CLASS_UNKNOWN, TAPI_EVENT_UNKNOWN, TAPI_REGISTRATION_FAILURE }

Index

- AccessType
  - TelSimFileAccessGeneralResp_t, 236
- AcInvalidate
  - TelSimFileInfo_t, 238
- Acl
  - TelSimApnData_t, 228
- AclLen
  - TelSimApnData_t, 228
- AcRead
  - TelSimFileInfo_t, 238
- AcRehabilitate
  - TelSimFileInfo_t, 238
- AcUpdate
  - TelSimFileInfo_t, 238
- AlertingPattern
  - TelCallIncomingCallInfo_t, 212
- AlphaIdLen
  - TelSmsParams_t, 272
- apnCount
  - TelSimDeviceContextData_t, 231
- ApnData
  - TelSimDeviceContextData_t, 232

- bAcl
  - TelUSimServiceTable_t, 294
- bAclEnable
  - TelSimDeviceContextData_t, 231
- bAdn
  - TelSimServiceTable_t, 256
- bAoc
  - TelSimServiceTable_t, 256
  - TelUSimServiceTable_t, 292
- BarringRecord
  - TelSsRecord_t, 283
- bAutoAnswerForeMLPP
  - TelSimServiceTable_t, 258
  - TelUSimServiceTable_t, 293
- bAutoSelection
  - TelNetworkConfigSettings_t, 220
- bBdn
  - TelSimServiceTable_t, 258
  - TelUSimServiceTable_t, 291
- bBdnEnable
  - TelSimDeviceContextData_t, 231
- bCallControl
  - TelSimServiceTable_t, 258
- bCallForwardingIndicationStatus
  - TelUSimServiceTable_t, 295
- bCallForwardingNumberPresent
  - TelSsForwardRecord_t, 281
- bCallHoldSupported
  - TelSsCapabilityInfo_t, 278
- bCBEnabled
  - TelSmsCbConfig_t, 268
- bCBSupported
  - TelSsCapabilityInfo_t, 278
- bCcp
  - TelSimServiceTable_t, 256
  - TelUSimServiceTable_t, 292
- bCellBroadCastMsgId
  - TelSimServiceTable_t, 257
- bCellBroadcastMsgId
  - TelUSimServiceTable_t, 292
- bCellBroadcastMsgIdRanges
  - TelSimServiceTable_t, 258
  - TelUSimServiceTable_t, 292
- bCFSupported
  - TelSsCapabilityInfo_t, 278
- bConferenceState
  - TelCallStatus_t, 217
- bCoOperativeNetworkList
  - TelSimServiceTable_t, 259
  - TelUSimServiceTable_t, 294
- bCpbcchiInformation
  - TelUSimServiceTable_t, 294
- bCPBCCHInfo
  - TelSimServiceTable_t, 260
- bCugFlag
  - TelCallCugInfo_t, 209
- bCWSupported
  - TelSsCapabilityInfo_t, 278
- bDePersonalizedControlKeys
  - TelSimServiceTable_t, 259
  - TelUSimServiceTable_t, 294
- bDisplayRplmn
  - TelNetworkInfo_t, 221
- bDisplaySpn
  - TelNetworkInfo_t, 221
- bECTSupported
  - TelSsCapabilityInfo_t, 278

bEnabledServiceTable
    TelUSimServiceTable_t, 294
bEnhancedMultilevelPrecedencePreemptionService
    TelSimServiceTable_t, 258
    TelUSimServiceTable_t, 293
bEquivalentPlmn
    TelNetworkSystemId_t, 225
bExtenedCapaConfigParams
    TelSimServiceTable_t, 260
bExtension1
    TelSimServiceTable_t, 257
bExtension2
    TelSimServiceTable_t, 257
    TelUSimServiceTable_t, 291
bExtension3
    TelSimServiceTable_t, 257
    TelUSimServiceTable_t, 291
bExtension4
    TelSimServiceTable_t, 259
    TelUSimServiceTable_t, 291
bExtension5
    TelUSimServiceTable_t, 295
bExtension8
    TelUSimServiceTable_t, 296
bFdn
    TelSimServiceTable_t, 256
    TelUSimServiceTable_t, 291
bFdnEnable
    TelSimDeviceContextData_t, 231
bGPRS
    TelSimServiceTable_t, 259
bGroupIdLevel1
    TelSimServiceTable_t, 257
    TelUSimServiceTable_t, 292
bGroupIdLevel2
    TelSimServiceTable_t, 257
    TelUSimServiceTable_t, 292
bGsmAccess
    TelUSimServiceTable_t, 293
bGsmSecurityContext
    TelUSimServiceTable_t, 294
bHome
    TelNetworkSystemId_t, 225
bHplmnSelectorAccessTech
    TelSimServiceTable_t, 260
    TelUSimServiceTable_t, 295
bIciIct
    TelUSimServiceTable_t, 292
bImage
    TelSimServiceTable_t, 259
    TelUSimServiceTable_t, 293
bImsiInvalidated
    TelSimDeviceContextData_t, 231
bInvestigationScan
    TelSimServiceTable_t, 260
    TelUSimServiceTable_t, 294
bIsDeviceCtxdataValid
    TelSimDeviceContextData_t, 231
blnd
    TelSimServiceTable_t, 257
bLocalPhoneBook
    TelUSimServiceTable_t, 291
bMailBoxDiallingNumbers
    TelUSimServiceTable_t, 295
bMenuSelection
    TelSimServiceTable_t, 258
bMexe
    TelSimServiceTable_t, 260
    TelUSimServiceTable_t, 295
bMms
    TelUSimServiceTable_t, 296
bMmsUserConnectivityParameters
    TelUSimServiceTable_t, 296
bMncIncludesPcsDigit
    TelNetworkSystemId_t, 225
bMoCall
    TelCallStatus_t, 217
bMoreDataPresent
    TelCallUusInfo_t, 219
bMoSMSControl
    TelSimServiceTable_t, 259
bMPTYSupported
    TelSsCapabilityInfo_t, 278
bMsgWaitingIndStatus
    TelUSimServiceTable_t, 295
bMsisdn
    TelSimServiceTable_t, 256
    TelUSimServiceTable_t, 293
bNetworkAlertIndInMS
    TelSimServiceTable_t, 259
bNetworkDlsAvail
    TelNetworkSystemId_t, 225
bNetworkIdentityAvail
    TelNetworkSystemId_t, 225
bNetworkTimeAvail
    TelNetworkSystemId_t, 225
bNetworkTimeZoneAvail
    TelNetworkSystemId_t, 225
bNia
    TelUSimServiceTable_t, 296
bOCIOCT
    TelUSimServiceTable_t, 292
bOperatorControlledPlmnSelectorAccessTech
    TelSimServiceTable_t, 260
    TelUSimServiceTable_t, 295
bOperatorPlmnList
    TelUSimServiceTable_t, 295
bPIN1DisableFunction

TelSimServiceTable_t, 256
bPlmnForbidden
  TelNetworkSystemId_t, 224
bPlmnNetworkName
  TelUSimServiceTable_t, 295
bPlmnSelector
  TelSimServiceTable_t, 256
bPowerOnGprsAttach
  TelNetworkConfigSettings_t, 220
bProactiveSimCommand
  TelSimServiceTable_t, 258
bRfu1
  TelSimServiceTable_t, 256
  TelUSimServiceTable_t, 293
bRfu2
  TelSimServiceTable_t, 257
bRoaming
  TelNetworkInfo_t, 221
bRplmnLastUsedAccessTech
  TelSimServiceTable_t, 260
  TelUSimServiceTable_t, 295
bRunAtCommand
  TelSimServiceTable_t, 259
  TelUSimServiceTable_t, 294
bSdn
  TelSimServiceTable_t, 257
  TelUSimServiceTable_t, 291
bServiceProviderDisplayInfo
  TelUSimServiceTable_t, 295
bShallBeSetToOne
  TelUSimServiceTable_t, 294
bSms
  TelSimServiceTable_t, 256
  TelUSimServiceTable_t, 292
bSmsCbdDataDownload
  TelSimServiceTable_t, 258
  TelUSimServiceTable_t, 293
bSMSP
  TelUSimServiceTable_t, 292
bSmsParams
  TelSimServiceTable_t, 257
bSmsPpDataDownload
  TelSimServiceTable_t, 258
  TelUSimServiceTable_t, 293
bSmsr
  TelSimServiceTable_t, 259
  TelUSimServiceTable_t, 292
bSolsa
  TelSimServiceTable_t, 259
  TelUSimServiceTable_t, 293
bSpn
  TelSimServiceTable_t, 257
  TelUSimServiceTable_t, 293
bUsed
  TelUsimEccInfo_t, 289
bUserControlledPLMNSelectorAccessTech
  TelSimServiceTable_t, 260
  TelUSimServiceTable_t, 293
bUsimCallControl
  TelUSimServiceTable_t, 294
bUsimGpsCallControl
  TelUSimServiceTable_t, 296
bUsimMoSmsControl
  TelUSimServiceTable_t, 294
bUSSDSupportedInCallControl
  TelSimServiceTable_t, 259
bValid
  TelSimFileInfo_t, 238
  TelSimImsiInfo_t, 242
bVbsGroupId
  TelUSimServiceTable_t, 296
bVbsGroupIDList
  TelSimServiceTable_t, 258
bVgcsGroupIdList
  TelSimServiceTable_t, 258
bVgcsIdList
  TelUSimServiceTable_t, 296

Call Data Structures, 165
CallDS
  TAPI_CALL_ALERTING_PATTERN_1, 167
  TAPI_CALL_ALERTING_PATTERN_2, 167
  TAPI_CALL_ALERTING_PATTERN_3, 167
  TAPI_CALL_ALERTING_PATTERN_4, 167
  TAPI_CALL_ALERTING_PATTERN_5, 167
  TAPI_CALL_ALERTING_PATTERN_6, 167
  TAPI_CALL_ALERTING_PATTERN_7, 167
  TAPI_CALL_ALERTING_PATTERN_8, 167
  TAPI_CALL_ANSWER_ACCEPT, 168
  TAPI_CALL_ANSWER_REJECT, 168
  TAPI_CALL_CONNECT_ACTIVE, 167
  TAPI_CALL_CONNECT_HELD, 167
  TAPI_CALL_CONNECT_IDLE, 167
  TAPI_CALL_CUG_NO_INFO, 170
  TAPI_CALL_CUG_SUPRESS_OA, 170
  TAPI_CALL_CUG_SUPRESS_OA_AND_CUG, 170
  TAPI_CALL_CUG_SUPRESS_PRF_CUG, 170
  TAPI_CALL_IDENTITY_DEFAULT, 170
  TAPI_CALL_IDENTITY_HIDE, 170
  TAPI_CALL_IDENTITY_SHOW, 170
  TAPI_CALL_NAME_AVAIL, 171
  TAPI_CALL_NAME_AVAIL_RESTRICTED, 171
  TAPI_CALL_NAME_RESTRICTED, 171
  TAPI_CALL_NAME_UNAVAIL, 171

TAPI_CALL_NO_ALERTING_PATTERN, 167
TAPI_CALL_TYPE_DATA, 171
TAPI_CALL_TYPE_E911, 171
TAPI_CALL_TYPE_VOICE, 171
TAPI_CALL_UUS_1, 171
TAPI_CALL_UUS_NONE, 171
TAPI_CALL_UUS_PROTO_IA5_CHARS, 171
TAPI_CALL_UUS_PROTO_OSI_HIGHER_LAYER, 171
TAPI_CALL_UUS_PROTO_Q931, 171
TAPI_CALL_UUS_PROTO_SPECIFIC, 171
TAPI_CALL_UUS_PROTO_V120, 171
TAPI_CAUSE_ACCESS_CLASS_BLOCKED, 169
TAPI_CAUSE_ACCESS_NOT_ALLOWED, 170
TAPI_CAUSE_ACCESS_REJECTED, 170
TAPI_CAUSE_ACCESSDISCARD, 168
TAPI_CAUSE_ALERT_STOP, 169
TAPI_CAUSE_BARR_BYOPERATOR, 168
TAPI_CAUSE_BEARERMODE_NOTIMPL, 168
TAPI_CAUSE_BEARERMODE_RESTRICTED, 168
TAPI_CAUSE_BEARERMODE_UNAUTH, 168
TAPI_CAUSE_BEARERMODE_UNAVAIL, 168
TAPI_CAUSE_BUSY, 168
TAPI_CAUSE_CALL_BARRED, 170
TAPI_CAUSE_CALLMETER_EXPIRED, 168
TAPI_CAUSE_CALLNO_ERROR, 168
TAPI_CAUSE_CANNOT_IDENTIFY_CALL, 170
TAPI_CAUSE_CKTUNAVAIL, 168
TAPI_CAUSE_CLASSNOTSUPPORTED, 169
TAPI_CAUSE_CLIENT_END, 169
TAPI_CAUSE_CONGESTION, 168
TAPI_CAUSE_DATACODING_ERROR, 169
TAPI_CAUSE_DEACTIVATION, 170
TAPI_CAUSE_DESTIN_INCOMPAT, 168
TAPI_CAUSE_DTMF_NOSPEECH, 168
TAPI_CAUSE_DTMF_REJECTED, 168
TAPI_CAUSE_EMERGENCY_CALL_NOT_ALLOWED, 170
TAPI_CAUSE_FACILITY_NOTIMPL, 168
TAPI_CAUSE_FACILITY_NOTSUBSCRIBED, 168
TAPI_CAUSE_FADE, 169
TAPI_CAUSE_FATAL_ERROR, 170
TAPI_CAUSE_HOLD_FAIL, 169
TAPI_CAUSE_IDLE, 169
TAPI_CAUSE_INCOM_CALL, 169
TAPI_CAUSE_INCOMINGCUGCALLS_BARRED, 168
TAPI_CAUSE_INTERCEPT, 169
TAPI_CAUSE_INVAL_MSG, 169
TAPI_CAUSE_INVALID_IDENTITY, 170
TAPI_CAUSE_INVALNUM, 168
TAPI_CAUSE_LOWER_LAYER_FAILURE, 170
TAPI_CAUSE_MEMCAPEXCEEDED, 169
TAPI_CAUSE_MODIFY_NOT_PERMITTED, 169
TAPI_CAUSE_MODIFY_SAME_BEARER, 169
TAPI_CAUSE_MPTY_ERROR, 168
TAPI_CAUSE_NETWORK_SPECIFIC, 169
TAPI_CAUSE_NO_CIRCUIT_AVAIL, 168
TAPI_CAUSE_NO_SERVICE_HERE, 170
TAPI_CAUSE_NOANSWER, 168
TAPI_CAUSE_NONCUGMEMBER, 168
TAPI_CAUSE_NORMAL_UNSPECIFIED, 169
TAPI_CAUSE_NUMBER_ERROR, 168
TAPI_CAUSE_NUMBERCHANGED, 168
TAPI_CAUSE_NWFAILURE, 168
TAPI_CAUSE_OPERATIONUNAVAIL, 168
TAPI_CAUSE_OUTOFORDER, 168
TAPI_CAUSE_PROTOCOL_ERROR, 168
TAPI_CAUSE_PROTOCOLID_ERROR, 169
TAPI_CAUSE_QOSUNAVAIL, 168
TAPI_CAUSE_RADIO_LINK_FAILURE, 170
TAPI_CAUSE_RANDOM_ACCESS_FAILURE, 170
TAPI_CAUSE_REEST_NOT_ALLOWED, 170
TAPI_CAUSE_REJ_FAIL, 169
TAPI_CAUSE_REJ_SRVC_NOT_AVL, 169
TAPI_CAUSE_REJECT, 169
TAPI_CAUSE_REMOTE_CKTUNAVAIL, 169
TAPI_CAUSE_REORDER, 169
TAPI_CAUSE_REPLACENOTSUPPORTED, 169
TAPI_CAUSE_RESOURCEUNAVAIL, 169
TAPI_CAUSE_SERVICE_NOT_ALLOWED, 170
TAPI_CAUSE_SERVICE_NOT_AVAIL, 170
TAPI_CAUSE_SERVICE_NOT_WORKING, 170
TAPI_CAUSE_SERVICE_NOTIMPL, 169
TAPI_CAUSE_SERVICE_UNAVAIL, 169

TAPI_CAUSE_SERVICEID_ERROR, 169
TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_CALL, 170
TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_SS, 170
TAPI_CAUSE_SIM_CALL_CONTROL_MODIFIED, 170
TAPI_CAUSE_SIM_ERROR, 170
TAPI_CAUSE_SS_USSD_BUSY, 169
TAPI_CAUSE_SSERRINVALIDTI, 169
TAPI_CAUSE_SSERRINVALIDTIVALUE, 169
TAPI_CAUSE_SSERROR, 169
TAPI_CAUSE_SSERRSYNTAXERROR, 169
TAPI_CAUSE_SSINCOMPATIBLESTATE, 169
TAPI_CAUSE_SSNEGATIVEPASSWORDCHECK, 169
TAPI_CAUSE_SSPROTOCOLERROR, 169
TAPI_CAUSE_SSSENTASPHASE1, 169
TAPI_CAUSE_SUCCESS, 168
TAPI_CAUSE_TEMPFAILURE, 169
TAPI_CAUSE_TYPENOTSUPPORTED, 169
TAPI_CAUSE_UNACCEPTABLE_IDENTITY, 170
TAPI_CAUSE_UNKNOWN, 169
TAPI_CAUSE_UNKNOWN_IDENTITY, 170
TAPI_CAUSE_WRONGNUM, 169

CallDS
- TAPI_CALL_UUS_DATA_LEN_MAX, 167
- TAPI_CALLING_NAME_SIZE_MAX, 167
- TelCallActiveState_t, 167
- TelCallAlertingPattern_t, 167
- TelCallAnswerType_t, 167
- TelCallCause_t, 168
- TelCallCugOptions_t, 170
- TelCallIdentityMode_t, 170
- TelCallNameMode_t, 170
- TelCallType_t, 171
- TelCallUusProtocolType_t, 171
- TelCallUusType_t, 171

CallEndCause
- TelCallEndInfo_t, 210

CallEndTime
- TelCallEndInfo_t, 210

CallHandle
- TelCallIncomingCallInfo_t, 211
- TelCallUserDataInfo_t, 218

CallingNameInfo
- TelCallIncomingCallInfo_t, 211

CallServices, 27

CallStartTime
- TelCallEndInfo_t, 210

CallState
- TelCallStatus_t, 217

CallType
- TelCallIncomingCallInfo_t, 211
- TelCallSetupParams_t, 215
- TelCallStatus_t, 217
- TelSsCallBarringInfo_t, 275

CardStatus
- TelSimStatusData_t, 262

CardType
- TelSimStatusData_t, 262

CbMsgType
- TelSmsCbMsg_t, 269

CliPresentationIndicator
- TelCallIncomingCallInfo_t, 211

CliRecord
- TelSsRecord_t, 283

CliStatus
- TelSsCliRecord_t, 279

Common, 151
- TAPI_EVENT_CALL_ALERT_IND, 153
- TAPI_EVENT_CALL_ANSWER_CNF, 154
- TAPI_EVENT_CALL_CONNECTED_IND, 153
- TAPI_EVENT_CALL_CONNECTING_IND, 153
- TAPI_EVENT_CALL_END_IND, 154
- TAPI_EVENT_CALL_FORWARD_IND, 154
- TAPI_EVENT_CALL_HOLD_CNF, 154
- TAPI_EVENT_CALL_HOLD_IND, 154
- TAPI_EVENT_CALL_INCOM_IND, 153
- TAPI_EVENT_CALL_RELEASE_ALL_CNF, 154
- TAPI_EVENT_CALL_RELEASE_CNF, 154
- TAPI_EVENT_CALL_RETRIEVE_CNF, 154
- TAPI_EVENT_CALL_RETRIEVE_IND, 154
- TAPI_EVENT_CALL_SEND_DTMF_CNF, 154
- TAPI_EVENT_CALL_SETUP_CNF, 153
- TAPI_EVENT_CALL_SETUPCONFERENCE_CNF, 154
- TAPI_EVENT_CALL_SETUPCONFERENCE_IND, 154
- TAPI_EVENT_CALL_SPLITCONFERENCE_CNF, 154
- TAPI_EVENT_CALL_TRANSFER_CNF, 154
- TAPI_EVENT_CALL_TRANSFER_IND, 154
- TAPI_EVENT_CALL_USERINFO_IND, 154
- TAPI_EVENT_CALL_WAITING_IND, 154
- TAPI_EVENT_CLASS_ALL, 153
- TAPI_EVENT_CLASS_CALL_DATA, 153

TAPI_EVENT_CLASS_CALL_VOICE, 153
TAPI_EVENT_CLASS_ERR, 153
TAPI_EVENT_CLASS_NETTEXT, 153
TAPI_EVENT_CLASS_NETWORK, 153
TAPI_EVENT_CLASS_SIM, 153
TAPI_EVENT_CLASS_SS, 153
TAPI_EVENT_CLASS_UTIL, 153
TAPI_EVENT_ERR_RESERVED1_IND, 156
TAPI_EVENT_ERR_RESERVED2_IND, 156
TAPI_EVENT_ERR_RESERVED3_IND, 156
TAPI_EVENT_ERR_RESERVED4_IND, 156
TAPI_EVENT_ERR_RESERVED5_IND, 156
TAPI_EVENT_NETTEXT_CB_INCOM_IND, 155
TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF, 154
TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF, 154
TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF, 154
TAPI_EVENT_NETTEXT_GET_COUNT_CNF, 154
TAPI_EVENT_NETTEXT_GET_PARAM_CNF, 154
TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF, 154
TAPI_EVENT_NETTEXT_INCOM_IND, 154
TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND, 155
TAPI_EVENT_NETTEXT_READ_SMS_CNF, 154
TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF, 154
TAPI_EVENT_NETTEXT_SENTSTATUS_CNF, 154
TAPI_EVENT_NETTEXT_SET_REQUEST_CNF, 154
TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF, 155
TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF, 155
TAPI_EVENT_NETWORK_CHANGE_IND, 155
TAPI_EVENT_NETWORK_CONFIG_CNF, 155
TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF, 155
TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF, 155
TAPI_EVENT_NETWORK_RADIO_INFO_IND, 155
TAPI_EVENT_NETWORK_SEARCH_CNF, 155
TAPI_EVENT_NETWORK_SEARCH_IND, 155
TAPI_EVENT_NETWORK_SELECT_CNF, 155
TAPI_EVENT_NETWORK_SETNWBAND_CNF, 155
TAPI_EVENT_NETWORK_SETNWMODE_CNF, 155
TAPI_EVENT_NETWORK_SETNWORDER_CNF, 155
TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF, 155
TAPI_EVENT_NETWORK_SETSERVICEDOMAIN_CNF, 155
TAPI_EVENT_SIM_CHANGE_PIN_CNF, 156
TAPI_EVENT_SIM_DISABLE_FDN_CNF, 156
TAPI_EVENT_SIM_DISABLE_PIN_CNF, 156
TAPI_EVENT_SIM_ENABLE_FDN_CNF, 156
TAPI_EVENT_SIM_ENABLE_PIN_CNF, 156
TAPI_EVENT_SIM_FDN_STATUS_IND, 156
TAPI_EVENT_SIM_FILE_INFO_CNF, 156
TAPI_EVENT_SIM_PB_ACCESS_READ_CNF, 156
TAPI_EVENT_SIM_PB_DELETE_CNF, 156
TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF, 156
TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF, 156
TAPI_EVENT_SIM_PB_UPDATE_CNF, 156
TAPI_EVENT_SIM_READ_FILE_CNF, 156
TAPI_EVENT_SIM_STATUS_IND, 156
TAPI_EVENT_SIM_UNBLOCK_PIN_CNF, 156
TAPI_EVENT_SIM_UPDATE_FILE_CNF, 156
TAPI_EVENT_SIM_VERIFY_PIN_CNF, 156
TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF, 155
TAPI_EVENT_SS_BARRING_CNF, 155
TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF, 155
TAPI_EVENT_SS_FORWARD_CNF, 155
TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF, 155

    TAPI_EVENT_SS_GET_PASSWORD_IND, 156
    TAPI_EVENT_SS_QUERYCAPABILITY_INFO_CNF, 156
    TAPI_EVENT_SS_USSD_CNF, 156
    TAPI_EVENT_SS_USSD_IND, 156
    TAPI_EVENT_SS_USSD_IND_NOTIFY, 156
    TAPI_EVENT_SS_USSD_IND_RSP, 156
    TAPI_EVENT_SS_USSDCNF_NOTIFY, 156
    TAPI_EVENT_SS_USSDCNF_RSP, 156
    TAPI_EVENT_SS_VERIFY_PWD_CNF, 156
    TAPI_EVENT_SS_WAITING_CNF, 155
    TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF, 155
    TAPI_EVENT_TUNNEL_INFO_CNF, 156
    TAPI_EVENT_TUNNEL_INFO_IND, 156
    TelAppCallback, 152
    TelTapiEventClass_t, 153
    TelTapiEventType_t, 153
ComponentOverview.h, 297
Condition
    TelSsForwardInfo_t, 280

Data
    TelSimReadFileData_t, 252
DataLen
    TelSimCallbackEvent_t, 229
    TelSimEvent_t, 235
    TelSimReadFileData_t, 252
    TelSimUpdateFileRequestData_t, 264
DataLength
    TelSimReadFileRequestData_t, 254
Day
    TelNetworkTimeAndZone_t, 226
DaylightSave
    TelNetworkTimeAndZone_t, 227
Dcs
    TelSsUssdMsgInfo_t, 284
DialNumLen
    TelSmsAddressInfo_t, 267

EccEmergencyServiceInfo
    TelUsimEccInfo_t, 289
EccInfo
    TelSimEccData_t, 233
EccLen
    TelUsimEccInfo_t, 289
EccRecord
    TelSimStatusData_t, 262
EccRecordCount
    TelSimStatusData_t, 262
EventClass
    TelTapiEvent_t, 286
EventStatus
    TelSimCallbackEvent_t, 229
    TelSimEvent_t, 235
EventType
    TelSimCallbackEvent_t, 229
    TelTapiEvent_t, 286

FileID
    TelSimReadFileRequestData_t, 253
FileId
    TelSimFileAccessGeneralResp_t, 236
    TelSimFileInfo_t, 238
    TelSimReadFileData_t, 252
    TelSimUpdateFileRequestData_t, 264
FileName
    TelSimFileID_t, 237
FileSize
    TelSimFileInfo_t, 239
FileType
    TelSimFileInfo_t, 238
    TelSimReadFileRequestData_t, 253
    TelSimUpdateFileRequestData_t, 264
Flavour
    TelSsCallBarringRecord_t, 276
ForwardCondition
    TelSsForwardRecord_t, 281
ForwardingRecord
    TelSsRecord_t, 283

Gid1Info
    TelSimDeviceContextData_t, 232
Gid2Info
    TelSimDeviceContextData_t, 232
GroupIdentifierLen
    TelSimGroupIdentifierInfo_t, 240

Hour
    TelNetworkTimeAndZone_t, 226

Iccid
    TelSimIccIdInfo_t, 241
    TelSimStatusData_t, 263
IccidLen
    TelSimStatusData_t, 263
IccidLength
    TelSimIccIdInfo_t, 241
IdentificationType
    TelSsCliRecord_t, 279
IdentityMode
    TelCallSetupParams_t, 215
Index
    TelCallCugInfo_t, 209
    TelSimPbRecordData_t, 247

IndexList
  TelSmsStoredMsgCountInfo_t, 274
IndType
  TelSsUssdMsgInfo_t, 284
ITapiCall.h, 298
ITapiNetText.h, 300
ITapiNetwork.h, 302
ITapiSim.h, 304
ITapiSs.h, 307

Length
  TelSmsCbMsg_t, 269
  TelTapiErrorInfo_t, 285
lp
  TelSimLanguageInfo_t, 243
LpCount
  TelSimLanguageInfo_t, 243
LpInfo
  TelSimStatusData_t, 263

MajorVersion
  TelTapiVersionInfo_t, 288
MinorVersion
  TelTapiVersionInfo_t, 288
Minute
  TelNetworkTimeAndZone_t, 226
Mode
  TelSsCallBarringInfo_t, 275
  TelSsForwardInfo_t, 280
Month
  TelNetworkTimeAndZone_t, 226
MsgIdCount
  TelSmsCbConfig_t, 268
MsgIDs
  TelSmsCbConfig_t, 268
MsgLength
  TelSmsDatapackageInfo_t, 271
MsgStatus
  TelSmsData_t, 270

Name
  TelSimPb2gData_t, 244
NameEncryptType
  TelSimPb2gData_t, 244
NameLen
  TelSimPb2gData_t, 244
NameMode
  TelCallingNameInfo_t, 213
NetText Data Structures, 172
NetTextDS
  TAPI_NETTEXT_BEARER_CS_ONLY, 174
  TAPI_NETTEXT_BEARER_CS_PREFERRED, 174
  TAPI_NETTEXT_BEARER_PS_ONLY, 174
  TAPI_NETTEXT_BEARER_PS_PREFERRED, 174
  TAPI_NETTEXT_CALL_BARRED, 176
  TAPI_NETTEXT_CB_MSG_CBS, 175
  TAPI_NETTEXT_CB_MSG_CBS41, 175
  TAPI_NETTEXT_CB_MSG_INVALID, 175
  TAPI_NETTEXT_CB_MSG_SCHEDULE, 175
  TAPI_NETTEXT_CONGESTION, 176
  TAPI_NETTEXT_DEST_ADDRESS_FDN_RESTRICTED, 175
  TAPI_NETTEXT_DESTINAITION_OUTOFSERVICE, 176
  TAPI_NETTEXT_DEVICE_FAILURE, 175
  TAPI_NETTEXT_FACILITY_NOT_IMPLEMENTED, 176
  TAPI_NETTEXT_FACILITY_NOT_SUBSCRIBED, 176
  TAPI_NETTEXT_FACILITY_REJECTED, 176
  TAPI_NETTEXT_IE_NOT_IMPLEMENTED, 176
  TAPI_NETTEXT_INTERWORKING, 176
  TAPI_NETTEXT_INVALID_LOCATION, 175
  TAPI_NETTEXT_INVALID_MANDATORY_INFO, 176
  TAPI_NETTEXT_INVALID_MSG, 176
  TAPI_NETTEXT_INVALID_PARAMETER, 175
  TAPI_NETTEXT_INVALID_PARAMETER_FORMAT, 174
  TAPI_NETTEXT_INVALID_REF_VALUE, 176
  TAPI_NETTEXT_MEMORY_CAPACITY_EXCEEDED, 176
  TAPI_NETTEXT_MEMORY_FAILURE, 174
  TAPI_NETTEXT_MESSAGE_NOT_COMPAT_PROTOCOL, 176
  TAPI_NETTEXT_MESSAGE_TRANSFER_REJECTED, 176
  TAPI_NETTEXT_MESSAGE_TYPE_NOT_IMPLEMENTED, 176
  TAPI_NETTEXT_NETWORK_OUTOFORDER, 176
  TAPI_NETTEXT_NO_NETWORK_RESP, 175
  TAPI_NETTEXT_NO_PREFERRED_BEARER, 174
  TAPI_NETTEXT_NO_SIM, 175
  TAPI_NETTEXT_OPERATION_NOT_SUPPORTED, 174
  TAPI_NETTEXT_OPERATOR_DETERMINED_BARRING, 176

  TAPI_NETTEXT_PHONE_FAILURE, 174
  TAPI_NETTEXT_PROTOCOL_ERROR, 176
  TAPI_NETTEXT_RESEND_ALREADY_DONE, 175
  TAPI_NETTEXT_RESOURCES_UNAVAILABLE, 176
  TAPI_NETTEXT_ROUTING_NOT_AVAILABLE, 175
  TAPI_NETTEXT_SCADDRESS_FDN_RESTRICTED, 175
  TAPI_NETTEXT_SCADDRESS_NOT_AVAILABLE, 175
  TAPI_NETTEXT_SENDSMS_SUCCESS, 175
  TAPI_NETTEXT_SERVICE_RESERVED, 175
  TAPI_NETTEXT_SIM_BUSY, 174
  TAPI_NETTEXT_SIM_FAILURE, 174
  TAPI_NETTEXT_SIM_NOT_READY, 175
  TAPI_NETTEXT_STATUS_DELIVERED, 175
  TAPI_NETTEXT_STATUS_DELIVERY_UNCONFIRMED, 175
  TAPI_NETTEXT_STATUS_MESSAGE_REPLACED, 175
  TAPI_NETTEXT_STATUS_READ, 175
  TAPI_NETTEXT_STATUS_RESERVED, 175
  TAPI_NETTEXT_STATUS_SENT, 175
  TAPI_NETTEXT_STATUS_UNREAD, 175
  TAPI_NETTEXT_STATUS_UNSENT, 175
  TAPI_NETTEXT_SUCCESS, 174
  TAPI_NETTEXT_TEMPORARY_FAILURE, 176
  TAPI_NETTEXT_UNASSIGNED_NUMBER, 176
  TAPI_NETTEXT_UNKNOWN, 174
  TAPI_NETTEXT_UNKNOWN_SUBSCRIBER, 176
  TAPI_NETTEXT_UNSPECIFIED_SUBSCRIBER, 176
NetTextDS
  TAPI_NETTEXT_CB_PAGE_SIZE_MAX, 173
  TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX, 173
  TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX, 173
  TAPI_NETTEXT_SMDATA_SIZE_MAX, 173
  TAPI_SIM_SMSP_ADDRESS_LEN, 174
  TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX, 174
  TelSmsBearerType_t, 174
  TelSmsCause_t, 174
  TelSmsCbMsgType_t, 174
  TelSmsMsgStatus_t, 175
  TelSmsResponse_t, 175
NetTextServices, 49
Network Data Structures, 177
NetworkDS
  TAPI_NETWORK_BAND_MODE_ONLY, 179
  TAPI_NETWORK_BAND_MODE_PREFERRED, 179
  TAPI_NETWORK_BAND_TYPE_ANY, 179
  TAPI_NETWORK_BAND_TYPE_GSM, 179
  TAPI_NETWORK_BAND_TYPE_GSM1900, 179
  TAPI_NETWORK_BAND_TYPE_GSM850, 179
  TAPI_NETWORK_BAND_TYPE_GSM_900_1800, 179
  TAPI_NETWORK_BAND_TYPE_WCDMA, 179
  TAPI_NETWORK_BAND_TYPE_WCDMA1900, 179
  TAPI_NETWORK_BAND_TYPE_WCDMA2100, 179
  TAPI_NETWORK_BAND_TYPE_WCDMA850, 179
  TAPI_NETWORK_CONFIG_SETTINGS_FAILURE_ERR, 180
  TAPI_NETWORK_DLS_ADJ_NO_ADJUSTMENT, 179
  TAPI_NETWORK_DLS_ADJ_PLUS_ONE_HOUR, 179
  TAPI_NETWORK_DLS_PLUS_TWO_HOUR, 179
  TAPI_NETWORK_INTERNAL_FAILURE_ERR, 180
  TAPI_NETWORK_MODE_AUTOMATIC, 179
  TAPI_NETWORK_MODE_CDMA, 179
  TAPI_NETWORK_MODE_EMERGENCY, 179
  TAPI_NETWORK_MODE_GSM, 179
  TAPI_NETWORK_MODE_WCDMA, 179
  TAPI_NETWORK_MODE_WLAN, 179
  TAPI_NETWORK_NO_ERR, 180
  TAPI_NETWORK_OFFLINE_ERR, 180
  TAPI_NETWORK_OPERATION_FAILED, 180
  TAPI_NETWORK_PHONE_IN_USE_ERR, 180
  TAPI_NETWORK_SELECTIONMODE_AUTOMATIC, 180
  TAPI_NETWORK_SELECTIONMODE_MANUAL, 180

TAPI_NETWORK_SERVICE_DOMAIN_AUTOMATIC, 180
TAPI_NETWORK_SERVICE_DOMAIN_COMBINED, 180
TAPI_NETWORK_SERVICE_DOMAIN_CS, 180
TAPI_NETWORK_SERVICE_DOMAIN_PS, 180
TAPI_NETWORK_SERVICE_LEVEL_EMERGENCY, 180
TAPI_NETWORK_SERVICE_LEVEL_FULL, 181
TAPI_NETWORK_SERVICE_LEVEL_NO, 180
TAPI_NETWORK_SERVICE_LEVEL_SEARCH, 181
TAPI_NETWORK_SERVICE_TYPE_COMBINED, 181
TAPI_NETWORK_SERVICE_TYPE_CS, 181
TAPI_NETWORK_SERVICE_TYPE_LIMITED, 181
TAPI_NETWORK_SERVICE_TYPE_NO_SVE, 181
TAPI_NETWORK_SERVICE_TYPE_PS, 181
TAPI_NETWORK_SYSID_FORMAT_PLMN, 181
TAPI_NETWORK_SYSID_FORMAT_SID, 181
TAPI_NETWORK_SYSTEM_CDMA, 181
TAPI_NETWORK_SYSTEM_EGPRS, 181
TAPI_NETWORK_SYSTEM_GPRS, 181
TAPI_NETWORK_SYSTEM_GSM, 181
TAPI_NETWORK_SYSTEM_GSM_AND_UMTS, 181
TAPI_NETWORK_SYSTEM_NO_SRV, 181
TAPI_NETWORK_SYSTEM_PCS1900, 181
TAPI_NETWORK_SYSTEM_TDMA, 181
TAPI_NETWORK_SYSTEM_UMTS, 181

NetworkDS
  TAPI_NETWORK_LIST_MAX, 178
  TAPI_NETWORK_NAME_LEN_MAX, 178
  TAPI_NETWORK_ORDER_MAX, 178
  TAPI_NETWORK_SPN_LEN_MAX, 178
  TelNetworkBand_t, 178
  TelNetworkBandPreferred_t, 179
  TelNetworkDayLightSavingAdj_t, 179
  TelNetworkMode_t, 179
  TelNetworkOperationCause_t, 179
  TelNetworkSelectionMode_t, 180
  TelNetworkServiceDomain_t, 180
  TelNetworkServiceLevel_t, 180
  TelNetworkServiceType_t, 181
  TelNetworkSystemIdFormat_t, 181
  TelNetworkSystemType_t, 181
NetworkMode
  TelNetworkConfigSettings_t, 220
  TelNetworkSystemId_t, 224
NetworkOrder
  TelNetworkConfigSettings_t, 220
NetworkServices, 69
NetworkTimeZone
  TelNetworkSystemId_t, 225
NoOfNwModePresent
  TelNetworkConfigSettings_t, 220
Npi
  TelSmsAddressInfo_t, 267
Number
  TelSimPb2gData_t, 244
NumberOfRecords
  TelSsInfo_t, 282
NumLen
  TelSimPb2gData_t, 244
NumOfNwAvailable
  TelNetworkList_t, 222
NumType
  TelSimPb2gData_t, 244

Offset
  TelSimReadFileRequestData_t, 253
  TelSimUpdateFileRequestData_t, 264
Option
  TelCallCugInfo_t, 209

ParamIndicator
  TelSmsParams_t, 272
PbAccessResult
  TelSimPbReadResp_t, 246
  TelSimPbUpdateResp_t, 249
PbIndex
  TelSimPbUpdateResp_t, 249
PbIndexMax
  TelSimPbEntryInfo_t, 245
PbIndexMin
  TelSimPbEntryInfo_t, 245
PbNumLenMax
  TelSimPbEntryInfo_t, 245
PbStorageData
  TelSimPbReadResp_t, 246
PbStorageInfo
  TelSimPbUpdateResp_t, 249
PbTextLenMax
  TelSimPbEntryInfo_t, 245
pCallHandle
  TelCallEndInfo_t, 210
pCugInfo
  TelCallSetupParams_t, 215

pData
    TelSimCallbackEvent_t, 229
    TelSimEvent_t, 235
    TelTapiEvent_t, 286
pErrorData
    TelTapiErrorInfo_t, 285
Pin1Status
    TelSimStatusData_t, 262
Pin2Status
    TelSimStatusData_t, 262
PinType
    TelSimPinData_t, 250
    TelSimPinInfo_t, 251
pNumber
    TelCallStatus_t, 217
pPbData
    TelSimPbRecordData_t, 247
ProtocolType
    TelCallUusInfo_t, 219
pSCA
    TelSmsDatapackageInfo_t, 271
pTunnelData
    TelTapiTunnelInfo_t, 287
pUusInfo
    TelCallSetupParams_t, 215

RecordCount
    TelSimFileInfo_t, 239
RecordIndex
    TelSimReadFileRequestData_t, 253
    TelSimUpdateFileRequestData_t, 264
    TelSmsParams_t, 272
RecordIndexFrom
    TelSimReadFileRequestData_t, 253
RecordIndexTo
    TelSimReadFileRequestData_t, 253
RecordLen
    TelSimFileInfo_t, 239
    TelSmsParams_t, 272
RecordMode
    TelSimReadFileRequestData_t, 253
    TelSimUpdateFileRequestData_t, 264
RedirectInfo
    TelCallIncomingCallInfo_t, 212
RemainingAttempts
    TelSimPinInfo_t, 251
RequestId
    TelSimCallbackEvent_t, 229
    TelSimEvent_t, 235
    TelTapiEvent_t, 286
RssiBar
    TelNetworkRadioInfo_t, 223
RxLevel
    TelNetworkRadioInfo_t, 223
RxQuality
    TelNetworkRadioInfo_t, 223

Second
    TelNetworkTimeAndZone_t, 226
SelectedId
    TelSmsCbConfig_t, 268
ServiceDomain
    TelNetworkConfigSettings_t, 220
ServiceLevel
    TelNetworkInfo_t, 221
ServiceType
    TelNetworkSystemId_t, 224
Signal
    TelCallIncomingCallInfo_t, 212
SIM Data Structures, 182
SimDeviceCtxData
    TelSimDeviceContext_t, 230
SIMDS
    TAPI_SIM_ACCESS_ACCESS_CONDITION_NOT_SATISFIED, 189
    TAPI_SIM_ACCESS_CARD_ERROR, 189
    TAPI_SIM_ACCESS_FAILED, 189
    TAPI_SIM_ACCESS_FILE_NOT_FOUND, 189
    TAPI_SIM_ACCESS_SUCCESS, 189
    TAPI_SIM_ACL_ADM, 189
    TAPI_SIM_ACL_ALW, 189
    TAPI_SIM_ACL_MAX_LEN, 187
    TAPI_SIM_ACL_NEVER, 189
    TAPI_SIM_ACL_PIN1, 189
    TAPI_SIM_ACL_PIN2, 189
    TAPI_SIM_ACL_RFU, 189
    TAPI_SIM_BUSY_STATUS, 196
    TAPI_SIM_CARD_ERROR_STATUS, 196
    TAPI_SIM_CARD_TYPE_GSM, 190
    TAPI_SIM_CARD_TYPE_UNKNOWN, 190
    TAPI_SIM_CARD_TYPE_USIM, 190
    TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX, 187
    TAPI_SIM_ECC_CODE_LEN, 187
    TAPI_SIM_ECC_ESC_AMBULANCE, 190
    TAPI_SIM_ECC_ESC_FIREBRIGADE, 190
    TAPI_SIM_ECC_ESC_MARAINEGUARD, 190
    TAPI_SIM_ECC_ESC_MOUTAINRESCUE, 190
    TAPI_SIM_ECC_ESC_POLICE, 190
    TAPI_SIM_ECC_ESC_SPARE, 190
    TAPI_SIM_ECC_MAX_LEN, 187
    TAPI_SIM_EFILE_ACM, 191
    TAPI_SIM_EFILE_ACM_MAX, 191
    TAPI_SIM_EFILE_ADN, 190
    TAPI_SIM_EFILE_BDN, 191

TAPI_SIM_EFILE_CBMI, 191
TAPI_SIM_EFILE_CCP, 191
TAPI_SIM_EFILE_DIR, 190
TAPI_SIM_EFILE_ECC, 191
TAPI_SIM_EFILE_EST, 191
TAPI_SIM_EFILE_EXT1, 191
TAPI_SIM_EFILE_EXT2, 191
TAPI_SIM_EFILE_EXT3, 191
TAPI_SIM_EFILE_EXT4, 191
TAPI_SIM_EFILE_FDN, 190
TAPI_SIM_EFILE_GID1, 191
TAPI_SIM_EFILE_GID2, 191
TAPI_SIM_EFILE_ICCID, 190
TAPI_SIM_EFILE_IMG, 191
TAPI_SIM_EFILE_IMSI, 191
TAPI_SIM_EFILE_INVALID, 193
TAPI_SIM_EFILE_LP, 191
TAPI_SIM_EFILE_MBDN, 191
TAPI_SIM_EFILE_MSISDN, 191
TAPI_SIM_EFILE_OTHERS, 193
TAPI_SIM_EFILE_PHASE, 191
TAPI_SIM_EFILE_PNN, 191
TAPI_SIM_EFILE_PUCT, 191
TAPI_SIM_EFILE_SDN, 191
TAPI_SIM_EFILE_SMS, 191
TAPI_SIM_EFILE_SMSP, 191
TAPI_SIM_EFILE_SMSS, 191
TAPI_SIM_EFILE_SPN, 191
TAPI_SIM_EFILE_SST, 191
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_AAS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRC, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CC, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CCP1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EXT1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GAS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR2, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PSC, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PUID, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE1, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID1, 193
TAPI_SIM_EFILE_USIM_ACM, 191
TAPI_SIM_EFILE_USIM_ACM_MAX, 191
TAPI_SIM_EFILE_USIM_BDN, 192
TAPI_SIM_EFILE_USIM_CBMI, 191
TAPI_SIM_EFILE_USIM_CCP2, 192
TAPI_SIM_EFILE_USIM_ECC, 191
TAPI_SIM_EFILE_USIM_EXT1, 192
TAPI_SIM_EFILE_USIM_EXT2, 192
TAPI_SIM_EFILE_USIM_EXT3, 192
TAPI_SIM_EFILE_USIM_EXT4, 192
TAPI_SIM_EFILE_USIM_EXT5, 192
TAPI_SIM_EFILE_USIM_FDN, 191
TAPI_SIM_EFILE_USIM_HIDDEN_KEY, 192
TAPI_SIM_EFILE_USIM_IMSI, 191
TAPI_SIM_EFILE_USIM_LI, 191

TAPI_SIM_EFILE_USIM_MBDN, 192
TAPI_SIM_EFILE_USIM_MBI, 192
TAPI_SIM_EFILE_USIM_MSISDN, 191
TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_CC, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_CCP1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL1, 193
TAPI_SIM_EFILE_USIM_PHONEBOOK_EXT1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP1, 193
TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PBR, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PSC, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PUID, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE1, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_UID, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_UID1, 192
TAPI_SIM_EFILE_USIM_PUCT, 191
TAPI_SIM_EFILE_USIM_SDN, 192
TAPI_SIM_EFILE_USIM_SMS, 191
TAPI_SIM_EFILE_USIM_SMSP, 191
TAPI_SIM_EFILE_USIM_SMSS, 191
TAPI_SIM_EFILE_USIM_SPN, 191
TAPI_SIM_EFILE_USIM_UST, 191
TAPI_SIM_FTYPE_CYCLIC, 194
TAPI_SIM_FTYPE_DEDICATED, 194
TAPI_SIM_FTYPE_INVALID_TYPE, 194
TAPI_SIM_FTYPE_LINEAR_FIXED, 194
TAPI_SIM_FTYPE_TRANSPARENT, 194
TAPI_SIM_GET_RESPONSE, 189
TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX, 187
TAPI_SIM_GSM_SST_SERVICE_MAX_COUNT, 187
TAPI_SIM_ICCID_MAX_LEN, 187
TAPI_SIM_ICCID_PACKET_LEN, 187
TAPI_SIM_ICCIDLENGTH, 187
TAPI_SIM_IMSI_LEN, 187
TAPI_SIM_IMSI_PACKET_LEN, 187
TAPI_SIM_INCOMPATIBLE_PIN_OPERATION_STATUS, 196
TAPI_SIM_LI_DANISH, 194
TAPI_SIM_LI_DUTCH, 194
TAPI_SIM_LI_ENGLISH, 194
TAPI_SIM_LI_FINNISH, 194
TAPI_SIM_LI_FRENCH, 194
TAPI_SIM_LI_GERMAN, 194
TAPI_SIM_LI_GREEK, 194
TAPI_SIM_LI_HUNGARIAN, 194
TAPI_SIM_LI_ITALIAN, 194
TAPI_SIM_LI_LANG_UNSPECIFIED, 194
TAPI_SIM_LI_MAX_LEN, 187
TAPI_SIM_LI_NORWEGIAN, 194
TAPI_SIM_LI_PACKET_MAX_LEN, 188
TAPI_SIM_LI_POLISH, 194
TAPI_SIM_LI_PORTUGUESE, 194
TAPI_SIM_LI_SPANISH, 194

TAPI_SIM_LI_SWEDISH, 194
TAPI_SIM_LI_TURKISH, 194
TAPI_SIM_LP_MAX_LEN, 188
TAPI_SIM_LP_PACKET_MAX_LEN, 188
TAPI_SIM_MCC_CODE_LEN, 188
TAPI_SIM_MNC_CODE_LEN, 188
TAPI_SIM_MSIN_CODE_LEN, 188
TAPI_SIM_NPI_DATA_NUMBERING_PLAN, 194
TAPI_SIM_NPI_ERMES_NUMBERING_PLAN, 195
TAPI_SIM_NPI_ISDN_TEL, 194
TAPI_SIM_NPI_NATIONAL, 195
TAPI_SIM_NPI_PRIVATE, 195
TAPI_SIM_NPI_RESERVED_FOR_EXT, 195
TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN, 194
TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN2, 194
TAPI_SIM_NPI_TELEX, 194
TAPI_SIM_NPI_UNKNOWN, 194
TAPI_SIM_NUM_TEXT_MAX_LEN, 188
TAPI_SIM_PB_ACCESS_CONDITION_NOT_SATISFIED, 195
TAPI_SIM_PB_ACCESS_MODE_ADD, 195
TAPI_SIM_PB_ACCESS_MODE_DEL, 195
TAPI_SIM_PB_ACCESS_MODE_EDIT, 195
TAPI_SIM_PB_ADN, 195
TAPI_SIM_PB_FAIL, 195
TAPI_SIM_PB_FDN, 195
TAPI_SIM_PB_INVALID_INDEX, 195
TAPI_SIM_PB_INVALID_NAME_LENGTH, 195
TAPI_SIM_PB_INVALID_NUMBER_LENGTH, 195
TAPI_SIM_PB_LDN, 195
TAPI_SIM_PB_MSISDN, 195
TAPI_SIM_PB_SDN, 195
TAPI_SIM_PB_SUCCESS, 195
TAPI_SIM_PIN2_INCORRECT_PASSWORD_STATUS, 196
TAPI_SIM_PIN2_REQD_STATUS, 196
TAPI_SIM_PIN_INCORRECT_PASSWORD_STATUS, 196
TAPI_SIM_PIN_LENGTH, 188
TAPI_SIM_PIN_OPERATION_SUCCESS, 196
TAPI_SIM_PIN_STATUS_BLOCKED, 196
TAPI_SIM_PIN_STATUS_DISABLED, 196
TAPI_SIM_PIN_STATUS_ENABLED, 196
TAPI_SIM_PIN_STATUS_PUK_BLOCKED, 196
TAPI_SIM_PIN_STATUS_UNKNOWN, 196
TAPI_SIM_PTYPE_ADM, 196
TAPI_SIM_PTYPE_PIN1, 196
TAPI_SIM_PTYPE_PIN2, 196
TAPI_SIM_PTYPE_PUK1, 196
TAPI_SIM_PTYPE_PUK2, 196
TAPI_SIM_PTYPE_UPIN, 196
TAPI_SIM_PUK2_REQD_STATUS, 196
TAPI_SIM_PUK_INCORRECT_PASSWORD_STATUS, 196
TAPI_SIM_PUK_REQD_STATUS, 196
TAPI_SIM_READ_BINARY, 189
TAPI_SIM_READ_RECORD, 189
TAPI_SIM_RECORD_ABSOLUTE, 197
TAPI_SIM_RECORD_CURRENT, 197
TAPI_SIM_RECORD_FROM_THE_FIRST, 197
TAPI_SIM_RECORD_NEXT, 197
TAPI_SIM_RECORD_NONE, 197
TAPI_SIM_RECORD_PREVIOUS, 197
TAPI_SIM_RECORD_TO_THE_LAST, 197
TAPI_SIM_SELECT, 189
TAPI_SIM_SST_MAX_LEN, 188
TAPI_SIM_STATUS, 189
TAPI_SIM_STATUS_CARD_BLOCKED, 190
TAPI_SIM_STATUS_CARD_ERROR, 190
TAPI_SIM_STATUS_CARD_NOT_PRESENT, 190
TAPI_SIM_STATUS_SIM_INIT_COMPLETED, 190
TAPI_SIM_STATUS_SIM_INITIALIZING, 190
TAPI_SIM_STATUS_SIM_PIN_REQUIRED, 190
TAPI_SIM_STATUS_SIM_PUK_REQUIRED, 190
TAPI_SIM_TEXT_ENC_ASCII, 197
TAPI_SIM_TEXT_ENC_GSM7BIT, 197
TAPI_SIM_TEXT_ENC_HEX, 197
TAPI_SIM_TEXT_ENC_UCS2, 197
TAPI_SIM_TON_ABBREVIATED_NUMBER, 197
TAPI_SIM_TON_ALPHA_NUMERIC, 197
TAPI_SIM_TON_DEDICATED_ACCESS, 197
TAPI_SIM_TON_INTERNATIONAL, 197
TAPI_SIM_TON_NATIONAL, 197
TAPI_SIM_TON_NETWORK_SPECIFIC, 197
TAPI_SIM_TON_RESERVED_FOR_EXT, 197
TAPI_SIM_TON_UNKNOWN, 197
TAPI_SIM_UECC_RECORD_COUNT_MAX, 188

    TAPI_SIM_UECC_RECORD_LEN_MAX, 188
    TAPI_SIM_UPDATE_BINARY, 189
    TAPI_SIM_UPDATE_RECORD, 189
    TAPI_TAPI_SIM_PBOOK_ENTRY_COUNT_MAX, 189
    TelSimAccessResult_t, 189
    TelSimAccessType_t, 189
    TelSimAcl_t, 189
    TelSimCardStatus_t, 189
    TelSimCardType_t, 190
    TelSimEccEmergencyServiceInfo_t, 190
    TelSimFileName_t, 190
    TelSimFileType_t, 193
    TelSimLangIndicationCode_t, 194
    TelSimNumberingPlanIdentity_t, 194
    TelSimPbAccessMode_t, 195
    TelSimPbAccessResult_t, 195
    TelSimPbFileType_t, 195
    TelSimPinOperationResult_t, 195
    TelSimPinStatus_t, 196
    TelSimPinType_t, 196
    TelSimRecordMode_t, 196
    TelSimTextEncrypt_t, 197
    TelSimTypeOfNum_t, 197
SimIndex
    TelSmsData_t, 270
SIMServices, 89
SimStatusData
    TelSimDeviceContext_t, 230
SmsData
    TelSmsData_t, 270
SS DataStructures, 198
SSDS
    TAPI_CALL_BARRING_ALL, 201
    TAPI_CALL_BARRING_ALL_INCOMING_CALLS, 201
    TAPI_CALL_BARRING_ALL_INCOMING_CALLS_INSIM, 201
    TAPI_CALL_BARRING_ALL_INCOMING_CALLS_ROAMING, 201
    TAPI_CALL_BARRING_ALL_OUTGOING_CALLS, 201
    TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL, 201
    TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL_EXCEPT, 201
    TAPI_CALL_BARRING_INCOMING, 201
    TAPI_CALL_BARRING_OUTGOING, 201
    TAPI_CALL_DIALDIGIT_LEN_MAX, 200
    TAPI_CALL_FORWARD_MODE_DISABLE_EV, 203
    TAPI_CALL_FORWARD_MODE_ENABLE_EV, 203
    TAPI_CALL_FORWARD_MODE_ERASURE_EV, 203
    TAPI_CALL_FORWARD_MODE_REGISTRATION_EV, 203
    TAPI_CALL_RESTRICT_ID_DEPEND_NETWORK_EV, 204
    TAPI_CALL_RESTRICT_ID_DISABLE_EV, 204
    TAPI_CALL_RESTRICT_ID_ENABLE_EV, 204
    TAPI_CALL_TYPE_DATA_CIRCUIT_SYNC_EV, 202
    TAPI_CALL_TYPE_DATA_EV, 201
    TAPI_CALL_TYPE_FAX_EV, 202
    TAPI_CALL_TYPE_SHORT_MESS_EV, 202
    TAPI_CALL_TYPE_VOICE_EV, 201
    TAPI_CALLED_LINE_IDENTITY_PRESENTATION, 204
    TAPI_CALLING_LINE_IDENTITY_PRESENTATION, 203
    TAPI_CALLING_LINE_IDENTITY_RESTRICTION, 203
    TAPI_CALLING_NAME_PRESENTATION, 204
    TAPI_CLI_STATUS_ACTIVATED, 203
    TAPI_CLI_STATUS_NOT_PROVISONED, 202
    TAPI_CLI_STATUS_PROVISIONED, 202
    TAPI_CLI_STATUS_TEMP_ALLOWED, 203
    TAPI_CLI_STATUS_TEMP_RESTRICTED, 203
    TAPI_CLI_STATUS_UNKOWN, 203
    TAPI_CONNECTED_LINE_IDENTITY_PRESENTATION, 204
    TAPI_CONNECTED_LINE_IDENTITY_RESTRICTION, 204
    TAPI_CS_FORWARD_TYPE_DATA_EV, 203
    TAPI_CS_FORWARD_TYPE_FAX_EV, 203
    TAPI_CS_FORWARD_TYPE_VOICE_EV, 203
    TAPI_SS_BAR_GET_PWD, 201
    TAPI_SS_BAR_NEW_PWD, 201
    TAPI_SS_BAR_VERIFY_NEW_PWD, 201
    TAPI_SS_BS_ALL_ASYNCHRONOUS, 205
    TAPI_SS_BS_ALL_BEARER_SERVICES, 205
    TAPI_SS_BS_ALL_DATA_CDS, 205
    TAPI_SS_BS_ALL_DATA_PADCA, 205
    TAPI_SS_BS_ALL_PLMN_SPEC_BS, 205

TAPI_SS_BS_ALL_SYNCHRONOUS, 205
TAPI_SS_BS_DATA_ALL_PDS, 205
TAPI_SS_BS_DATA_CIRCUIT_ASYNC, 205
TAPI_SS_BS_DATA_CIRCUIT_SYNC, 205
TAPI_SS_BS_PLMN_SPEC_TELE_1, 205
TAPI_SS_BS_PLMN_SPEC_TELE_2, 205
TAPI_SS_BS_PLMN_SPEC_TELE_3, 205
TAPI_SS_BS_PLMN_SPEC_TELE_4, 205
TAPI_SS_BS_PLMN_SPEC_TELE_5, 205
TAPI_SS_BS_PLMN_SPEC_TELE_6, 205
TAPI_SS_BS_PLMN_SPEC_TELE_7, 205
TAPI_SS_BS_PLMN_SPEC_TELE_8, 205
TAPI_SS_BS_PLMN_SPEC_TELE_9, 205
TAPI_SS_BS_PLMN_SPEC_TELE_A, 205
TAPI_SS_BS_PLMN_SPEC_TELE_B, 205
TAPI_SS_BS_PLMN_SPEC_TELE_C, 205
TAPI_SS_BS_PLMN_SPEC_TELE_D, 205
TAPI_SS_BS_PLMN_SPEC_TELE_E, 205
TAPI_SS_BS_PLMN_SPEC_TELE_F, 205
TAPI_SS_CALL_BAR_ACTIVATE, 201
TAPI_SS_CALL_BAR_DEACTIVATE, 201
TAPI_SS_CALLBARRED, 202
TAPI_SS_FACILITYNOTSUPPORTED, 202
TAPI_SS_FORWARD_WHEN_ALL_FORWARDING_EV, 203
TAPI_SS_FORWARD_WHEN_BUSY_EV, 203
TAPI_SS_FORWARD_WHEN_NO_ANSWER_EV, 203
TAPI_SS_FORWARD_WHEN_NOT_REACHABLE_EV, 203
TAPI_SS_FORWARD_WHEN_UNCONDITIONAL_EV, 203
TAPI_SS_GSM_BARR_PW_LEN_MAX, 200
TAPI_SS_ILLEGALSSOPERATION_ERROR, 202
TAPI_SS_INVALIDDEFLECTEDTONUMBER, 202
TAPI_SS_MAXNOMPTYEXCEEDED, 202
TAPI_SS_NEGATIVEPWCHECK, 202
TAPI_SS_NOTAVAILABLE, 202
TAPI_SS_NUMBEROFPWATTEMPTSVIOLATION, 202
TAPI_SS_OEM_NOT_SUPPORTED, 202
TAPI_SS_PWREGISTRATIONFAILURE, 202
TAPI_SS_REJECTEDBYNETWORK, 202
TAPI_SS_REJECTEDBYUSER, 202
TAPI_SS_RESOURCESNOTAVAILABLE, 202
TAPI_SS_SERVICENOTPROVISIONED, 202
TAPI_SS_STATUS_ACTIVE, 204
TAPI_SS_STATUS_NOTHING, 204
TAPI_SS_STATUS_PROVISIONED, 204
TAPI_SS_STATUS_QUIESCENT, 204
TAPI_SS_STATUS_REGISTERED, 204
TAPI_SS_SUBSCRIPTIONVIOLATION, 202
TAPI_SS_SUCCESS, 202
TAPI_SS_TIMEREXPIRE, 202
TAPI_SS_TS_ALL_DATA, 205
TAPI_SS_TS_ALL_DATA_TELESERVICES, 204
TAPI_SS_TS_ALL_PLMN_SPEC_TELE, 204
TAPI_SS_TS_ALL_SPEECH, 204
TAPI_SS_TS_ALL_TELE_AND_BEARER_SERVICES, 204
TAPI_SS_TS_ALL_TELESERVICES, 204
TAPI_SS_TS_ALL_TELESERVICES_EXCEPT_SMS, 204
TAPI_SS_TS_DATA_CIRCUIT_ASYNC, 205
TAPI_SS_TS_DATA_CIRCUIT_SYNC, 205
TAPI_SS_TS_DATA_PACKET, 205
TAPI_SS_TS_FACS, 204
TAPI_SS_TS_MULTIPLE, 205
TAPI_SS_TS_PLMN_SPEC_TELE_1, 204
TAPI_SS_TS_PLMN_SPEC_TELE_2, 205
TAPI_SS_TS_PLMN_SPEC_TELE_3, 205
TAPI_SS_TS_PLMN_SPEC_TELE_4, 205
TAPI_SS_TS_PLMN_SPEC_TELE_5, 205
TAPI_SS_TS_PLMN_SPEC_TELE_6, 205
TAPI_SS_TS_PLMN_SPEC_TELE_7, 205
TAPI_SS_TS_PLMN_SPEC_TELE_8, 205
TAPI_SS_TS_PLMN_SPEC_TELE_9, 205
TAPI_SS_TS_PLMN_SPEC_TELE_A, 205
TAPI_SS_TS_PLMN_SPEC_TELE_B, 205
TAPI_SS_TS_PLMN_SPEC_TELE_C, 205
TAPI_SS_TS_PLMN_SPEC_TELE_D, 205
TAPI_SS_TS_PLMN_SPEC_TELE_E, 205
TAPI_SS_TS_PLMN_SPEC_TELE_F, 205
TAPI_SS_TS_TELEPHONY, 205
TAPI_SS_TYPE_BARRING, 206
TAPI_SS_TYPE_CNAP, 206
TAPI_SS_TYPE_FORWARDING, 206
TAPI_SS_TYPE_IDENTIFICATION, 206
TAPI_SS_TYPE_WAITING, 206
TAPI_SS_UNEXPECTEDDATAVALUE, 202
TAPI_SS_UNKNOWNALPHABET, 202
TAPI_SS_UNKNOWNERROR, 202
TAPI_SS_UNKNOWNSUBSCRIBER, 202
TAPI_SS_USSD_DATA_SIZE_MAX, 200
TAPI_SS_USSDBUSY, 202
TAPI_USSD_IND_ACTION_REQUIRE, 206

- TAPI_USSD_IND_NO_ACTION_REQUIRE, 206
- TAPI_USSD_IND_NOT_SUPPORT, 206
- TAPI_USSD_IND_NOTIFY, 206
- TAPI_USSD_IND_OTHER_CLIENT, 206
- TAPI_USSD_IND_TERMINATED_BY_NET, 206
- TAPI_USSD_IND_TIME_OUT, 206
- TAPI_USSD_RES_DISPL_BUSY, 206
- TAPI_USSD_RES_NONE, 206
- TAPI_USSD_RES_OK, 206
- TAPI_USSD_RES_UNKNOWN_ALPHABET, 206
- TelSsBarPwdType_t, 201
- TelSsCallBarOperationMode_t, 201
- TelSsCallBarType_t, 201
- TelSsCallType_t, 201
- TelSsCause_t, 202
- TelSsCliStatus_t, 202
- TelSsForwardMode_t, 203
- TelSsForwardType_t, 203
- TelSsForwardWhen_t, 203
- TelSsLineIdentificationType_t, 203
- TelSsRestrictIDStatus_t, 204
- TelSsStatus_t, 204
- TelSsTeleService_t, 204
- TelSsType_t, 205
- TelSsUssdIndType_t, 206
- TelSsUssdRespType_t, 206
- TYPE_SS_TYPE_USSD, 206

SsRecord
- TelSsInfo_t, 282

SsStatus
- TelSsInfo_t, 282

Sst
- TelSimSstInfo_t, 261

SstData
- TelSimDeviceContextData_t, 232
- TelSimSstInfo_t, 261

SsTeleserviceType
- TelSsInfo_t, 282

SstInfo
- TelSimDeviceContextData_t, 232

SstLen
- TelSimSstInfo_t, 261

SsType
- TelSsInfo_t, 282

Status
- TelSsCallBarringRecord_t, 276
- TelSsCallWaitingRecord_t, 277
- TelSsForwardRecord_t, 281
- TelTapiEvent_t, 286

StorageFileType
- TelSimPbEntryInfo_t, 245
- TelSimPbRecordData_t, 247
- TelSimPbStorageInfo_t, 248

SupplementaryServices, 127

SysId
- TelNetworkInfo_t, 221
- TelNetworkSystemId_t, 224

SysIdFormat
- TelNetworkSystemId_t, 224

SystemList
- TelNetworkList_t, 222

SystemType
- TelNetworkSystemId_t, 224

szAlphaId
- TelSmsParams_t, 272

szCalledPartySubAddress
- TelCallSetupParams_t, 215

szCallForwardingNumber
- TelSsForwardRecord_t, 281

szCallingPartyNumber
- TelCallIncomingCallInfo_t, 211

szCallingPartySubAddress
- TelCallIncomingCallInfo_t, 211

szData
- TelSimUpdateFileRequestData_t, 264
- TelSmsDatapackageInfo_t, 271

szDiallingNum
- TelSmsAddressInfo_t, 267

szEcc
- TelUsimEccInfo_t, 289

szEcc1
- TelSimEccInfo_t, 234

szEcc2
- TelSimEccInfo_t, 234

szEcc3
- TelSimEccInfo_t, 234

szEcc4
- TelSimEccInfo_t, 234

szEcc5
- TelSimEccInfo_t, 234

szEccAlphaIdentifier
- TelUsimEccInfo_t, 289

szGroupIdentifier
- TelSimGroupIdentifierInfo_t, 240

szMcc
- TelSimImsiInfo_t, 242

szMnc
- TelSimImsiInfo_t, 242

szMsgData
- TelSmsCbMsg_t, 269

szMsin
- TelSimImsiInfo_t, 242

szNameData
- TelCallingNameInfo_t, 213

szNumber

TelCallSetupParams_t, 215
szPassword
TelSsCallBarringInfo_t, 275
szPath
TelSimFileID_t, 237
szPhoneNumber
TelSsForwardInfo_t, 280
szPinCode
TelSimPinData_t, 250
szRedirectedNumber
TelCallRedirectionInfo_t, 214
szRedirectSubAddress
TelCallRedirectionInfo_t, 214
szSpn
TelNetworkSystemId_t, 225
szSystemName
TelNetworkSystemId_t, 225
szUssdString
TelSsUssdMsgInfo_t, 284

TAPI_API_DEPRICATED
UtilityDS, 207
TAPI_API_INVALID_CALL_HANDLE
UtilityDS, 208
TAPI_API_INVALID_CALL_STATE
UtilityDS, 207
TAPI_API_INVALID_DATA_LEN
UtilityDS, 208
TAPI_API_INVALID_INPUT
UtilityDS, 207
TAPI_API_INVALID_PTR
UtilityDS, 207
TAPI_API_MODEM_POWERED_OFF
UtilityDS, 208
TAPI_API_NOT_SUPPORTED
UtilityDS, 207
TAPI_API_OEM_PLUGIN_FAILURE
UtilityDS, 208
TAPI_API_OFFLINE_MODE_ERROR
UtilityDS, 208
TAPI_API_REQUEST_MAX_IN_PROGRESS
UtilityDS, 208
TAPI_API_SERVER_FAILURE
UtilityDS, 208
TAPI_API_SERVICE_NOT_READY
UtilityDS, 208
TAPI_API_SIM_BUSY
UtilityDS, 208
TAPI_API_SIM_CARD_PERMANENT_ERROR
UtilityDS, 208
TAPI_API_SIM_INVALID_CARD_TYPE
UtilityDS, 208
TAPI_API_SIM_NOT_FOUND
UtilityDS, 208
TAPI_API_SIM_NOT_INITIALIZED
UtilityDS, 208
TAPI_API_SUCCESS
UtilityDS, 207
TAPI_API_SYSTEM_OUT_OF_MEM
UtilityDS, 207
TAPI_API_SYSTEM_RPC_LINK_DOWN
UtilityDS, 207
TAPI_CALL_ALERTING_PATTERN_1
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_2
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_3
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_4
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_5
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_6
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_7
CallDS, 167
TAPI_CALL_ALERTING_PATTERN_8
CallDS, 167
TAPI_CALL_ANSWER_ACCEPT
CallDS, 168
TAPI_CALL_ANSWER_REJECT
CallDS, 168
TAPI_CALL_BARRING_ALL
SSDS, 201
TAPI_CALL_BARRING_ALL_INCOMING_CALLS
SSDS, 201
TAPI_CALL_BARRING_ALL_INCOMING_CALLS_INSIM
SSDS, 201
TAPI_CALL_BARRING_ALL_INCOMING_CALLS_ROAMING
SSDS, 201
TAPI_CALL_BARRING_ALL_OUTGOING_CALLS
SSDS, 201
TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL
SSDS, 201
TAPI_CALL_BARRING_ALL_OUTGOING_INTERN_CALL_EXCEPT
SSDS, 201
TAPI_CALL_BARRING_INCOMING
SSDS, 201
TAPI_CALL_BARRING_OUTGOING
SSDS, 201
TAPI_CALL_CONNECT_ACTIVE
CallDS, 167

TAPI_CALL_CONNECT_HELD
  CallDS, 167
TAPI_CALL_CONNECT_IDLE
  CallDS, 167
TAPI_CALL_CUG_NO_INFO
  CallDS, 170
TAPI_CALL_CUG_SUPRESS_OA
  CallDS, 170
TAPI_CALL_CUG_SUPRESS_OA_AND_CUG
  CallDS, 170
TAPI_CALL_CUG_SUPRESS_PRF_CUG
  CallDS, 170
TAPI_CALL_DIALDIGIT_LEN_MAX
  SSDS, 200
TAPI_CALL_FORWARD_MODE_DISABLE_EV
  SSDS, 203
TAPI_CALL_FORWARD_MODE_ENABLE_EV
  SSDS, 203
TAPI_CALL_FORWARD_MODE_ERASURE_EV
  SSDS, 203
TAPI_CALL_FORWARD_MODE_REGISTRATION_EV
  SSDS, 203
TAPI_CALL_IDENTITY_DEFAULT
  CallDS, 170
TAPI_CALL_IDENTITY_HIDE
  CallDS, 170
TAPI_CALL_IDENTITY_SHOW
  CallDS, 170
TAPI_CALL_NAME_AVAIL
  CallDS, 171
TAPI_CALL_NAME_AVAIL_RESTRICTED
  CallDS, 171
TAPI_CALL_NAME_RESTRICTED
  CallDS, 171
TAPI_CALL_NAME_UNAVAIL
  CallDS, 171
TAPI_CALL_NO_ALERTING_PATTERN
  CallDS, 167
TAPI_CALL_RESTRICT_ID_DEPEND_NETWORK_EV
  SSDS, 204
TAPI_CALL_RESTRICT_ID_DISABLE_EV
  SSDS, 204
TAPI_CALL_RESTRICT_ID_ENABLE_EV
  SSDS, 204
TAPI_CALL_TYPE_DATA
  CallDS, 171
TAPI_CALL_TYPE_DATA_CIRCUIT_SYNC_EV
  SSDS, 202
TAPI_CALL_TYPE_DATA_EV
  SSDS, 201
TAPI_CALL_TYPE_E911
  CallDS, 171
TAPI_CALL_TYPE_FAX_EV
  SSDS, 202
TAPI_CALL_TYPE_SHORT_MESS_EV
  SSDS, 202
TAPI_CALL_TYPE_VOICE
  CallDS, 171
TAPI_CALL_TYPE_VOICE_EV
  SSDS, 201
TAPI_CALL_UUS_1
  CallDS, 171
TAPI_CALL_UUS_DATA_LEN_MAX
  CallDS, 167
TAPI_CALL_UUS_NONE
  CallDS, 171
TAPI_CALL_UUS_PROTO_IA5_CHARS
  CallDS, 171
TAPI_CALL_UUS_PROTO_OSI_HIGHER_LAYER
  CallDS, 171
TAPI_CALL_UUS_PROTO_Q931
  CallDS, 171
TAPI_CALL_UUS_PROTO_SPECIFIC
  CallDS, 171
TAPI_CALL_UUS_PROTO_V120
  CallDS, 171
TAPI_CALLED_LINE_IDENTITY_PRESENTATION
  SSDS, 204
TAPI_CALLING_LINE_IDENTITY_PRESENTATION
  SSDS, 203
TAPI_CALLING_LINE_IDENTITY_RESTRICTION
  SSDS, 203
TAPI_CALLING_NAME_PRESENTATION
  SSDS, 204
TAPI_CALLING_NAME_SIZE_MAX
  CallDS, 167
TAPI_CAUSE_ACCESS_CLASS_BLOCKED
  CallDS, 169
TAPI_CAUSE_ACCESS_NOT_ALLOWED
  CallDS, 170
TAPI_CAUSE_ACCESS_REJECTED
  CallDS, 170
TAPI_CAUSE_ACCESSDISCARD
  CallDS, 168
TAPI_CAUSE_ALERT_STOP
  CallDS, 169
TAPI_CAUSE_BARR_BYOPERATOR
  CallDS, 168
TAPI_CAUSE_BEARERMODE_NOTIMPL
  CallDS, 168

TAPI_CAUSE_BEARERMODE_RESTRICTED
  CallDS, 168
TAPI_CAUSE_BEARERMODE_UNAUTH
  CallDS, 168
TAPI_CAUSE_BEARERMODE_UNAVAIL
  CallDS, 168
TAPI_CAUSE_BUSY
  CallDS, 168
TAPI_CAUSE_CALL_BARRED
  CallDS, 170
TAPI_CAUSE_CALLMETER_EXPIRED
  CallDS, 168
TAPI_CAUSE_CALLNO_ERROR
  CallDS, 168
TAPI_CAUSE_CANNOT_IDENTIFY_CALL
  CallDS, 170
TAPI_CAUSE_CKTUNAVAIL
  CallDS, 168
TAPI_CAUSE_CLASSNOTSUPPORTED
  CallDS, 169
TAPI_CAUSE_CLIENT_END
  CallDS, 169
TAPI_CAUSE_CONGESTION
  CallDS, 168
TAPI_CAUSE_DATACODING_ERROR
  CallDS, 169
TAPI_CAUSE_DEACTIVATION
  CallDS, 170
TAPI_CAUSE_DESTIN_INCOMPAT
  CallDS, 168
TAPI_CAUSE_DTMF_NOSPEECH
  CallDS, 168
TAPI_CAUSE_DTMF_REJECTED
  CallDS, 168
TAPI_CAUSE_EMERGENCY_CALL_NOT_ALLOWED
  CallDS, 170
TAPI_CAUSE_FACILITY_NOTIMPL
  CallDS, 168
TAPI_CAUSE_FACILITY_NOTSUBSCRIBED
  CallDS, 168
TAPI_CAUSE_FADE
  CallDS, 169
TAPI_CAUSE_FATAL_ERROR
  CallDS, 170
TAPI_CAUSE_HOLD_FAIL
  CallDS, 169
TAPI_CAUSE_IDLE
  CallDS, 169
TAPI_CAUSE_INCOM_CALL
  CallDS, 169
TAPI_CAUSE_INCOMINGCUGCALLS_BARRED
  CallDS, 168
TAPI_CAUSE_INTERCEPT
  CallDS, 169
TAPI_CAUSE_INVAL_MSG
  CallDS, 169
TAPI_CAUSE_INVALID_IDENTITY
  CallDS, 170
TAPI_CAUSE_INVALNUM
  CallDS, 168
TAPI_CAUSE_LOWER_LAYER_FAILURE
  CallDS, 170
TAPI_CAUSE_MEMCAPEXCEEDED
  CallDS, 169
TAPI_CAUSE_MODIFY_NOT_PERMITTED
  CallDS, 169
TAPI_CAUSE_MODIFY_SAME_BEARER
  CallDS, 169
TAPI_CAUSE_MPTY_ERROR
  CallDS, 168
TAPI_CAUSE_NETWORK_SPECIFIC
  CallDS, 169
TAPI_CAUSE_NO_CIRCUIT_AVAIL
  CallDS, 168
TAPI_CAUSE_NO_SERVICE_HERE
  CallDS, 170
TAPI_CAUSE_NOANSWER
  CallDS, 168
TAPI_CAUSE_NONCUGMEMBER
  CallDS, 168
TAPI_CAUSE_NORMAL_UNSPECIFIED
  CallDS, 169
TAPI_CAUSE_NUMBER_ERROR
  CallDS, 168
TAPI_CAUSE_NUMBERCHANGED
  CallDS, 168
TAPI_CAUSE_NWFAILURE
  CallDS, 168
TAPI_CAUSE_OPERATIONUNAVAIL
  CallDS, 168
TAPI_CAUSE_OUTOFORDER
  CallDS, 168
TAPI_CAUSE_PROTOCOL_ERROR
  CallDS, 168
TAPI_CAUSE_PROTOCOLID_ERROR
  CallDS, 169
TAPI_CAUSE_QOSUNAVAIL
  CallDS, 168
TAPI_CAUSE_RADIO_LINK_FAILURE
  CallDS, 170
TAPI_CAUSE_RANDOM_ACCESS_FAILURE
  CallDS, 170
TAPI_CAUSE_REEST_NOT_ALLOWED
  CallDS, 170
TAPI_CAUSE_REJ_FAIL
  CallDS, 169

TAPI_CAUSE_REJ_SRVC_NOT_AVL
 CallDS, 169
TAPI_CAUSE_REJECT
 CallDS, 169
TAPI_CAUSE_REMOTE_CKTUNAVAIL
 CallDS, 169
TAPI_CAUSE_REORDER
 CallDS, 169
TAPI_CAUSE_REPLACENOTSUPPORTED
 CallDS, 169
TAPI_CAUSE_RESOURCEUNAVAIL
 CallDS, 169
TAPI_CAUSE_SERVICE_NOT_ALLOWED
 CallDS, 170
TAPI_CAUSE_SERVICE_NOT_AVAIL
 CallDS, 170
TAPI_CAUSE_SERVICE_NOT_WORKING
 CallDS, 170
TAPI_CAUSE_SERVICE_NOTIMPL
 CallDS, 169
TAPI_CAUSE_SERVICE_UNAVAIL
 CallDS, 169
TAPI_CAUSE_SERVICEID_ERROR
 CallDS, 169
TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_CALL
 CallDS, 170
TAPI_CAUSE_SIM_CALL_CONTROL_CHANGED_TO_SS
 CallDS, 170
TAPI_CAUSE_SIM_CALL_CONTROL_MODIFIED
 CallDS, 170
TAPI_CAUSE_SIM_ERROR
 CallDS, 170
TAPI_CAUSE_SS_USSD_BUSY
 CallDS, 169
TAPI_CAUSE_SSERRINVALIDTI
 CallDS, 169
TAPI_CAUSE_SSERRINVALIDTIVALUE
 CallDS, 169
TAPI_CAUSE_SSERROR
 CallDS, 169
TAPI_CAUSE_SSERRSYNTAXERROR
 CallDS, 169
TAPI_CAUSE_SSINCOMPATIBLESTATE
 CallDS, 169
TAPI_CAUSE_SSNEGATIVEPASSWORDCHECK
 CallDS, 169
TAPI_CAUSE_SSPROTOCOLERROR
 CallDS, 169
TAPI_CAUSE_SSSENTASPHASE1
 CallDS, 169
TAPI_CAUSE_SUCCESS
 CallDS, 168
TAPI_CAUSE_TEMPFAILURE
 CallDS, 169
TAPI_CAUSE_TYPENOTSUPPORTED
 CallDS, 169
TAPI_CAUSE_UNACCEPTABLE_IDENTITY
 CallDS, 170
TAPI_CAUSE_UNKNOWN
 CallDS, 169
TAPI_CAUSE_UNKNOWN_IDENTITY
 CallDS, 170
TAPI_CAUSE_WRONGNUM
 CallDS, 169
TAPI_CLI_STATUS_ACTIVATED
 SSDS, 203
TAPI_CLI_STATUS_NOT_PROVISONED
 SSDS, 202
TAPI_CLI_STATUS_PROVISIONED
 SSDS, 202
TAPI_CLI_STATUS_TEMP_ALLOWED
 SSDS, 203
TAPI_CLI_STATUS_TEMP_RESTRICTED
 SSDS, 203
TAPI_CLI_STATUS_UNKOWN
 SSDS, 203
TAPI_CONNECTED_LINE_IDENTITY_PRESENTATION
 SSDS, 204
TAPI_CONNECTED_LINE_IDENTITY_RESTRICTION
 SSDS, 204
TAPI_CS_FORWARD_TYPE_DATA_EV
 SSDS, 203
TAPI_CS_FORWARD_TYPE_FAX_EV
 SSDS, 203
TAPI_CS_FORWARD_TYPE_VOICE_EV
 SSDS, 203
TAPI_EVENT_CALL_ALERT_IND
 Common, 153
TAPI_EVENT_CALL_ANSWER_CNF
 Common, 154
TAPI_EVENT_CALL_CONNECTED_IND
 Common, 153
TAPI_EVENT_CALL_CONNECTING_IND
 Common, 153
TAPI_EVENT_CALL_END_IND
 Common, 154
TAPI_EVENT_CALL_FORWARD_IND
 Common, 154
TAPI_EVENT_CALL_HOLD_CNF
 Common, 154
TAPI_EVENT_CALL_HOLD_IND
 Common, 154
TAPI_EVENT_CALL_INCOM_IND

Common, 153
TAPI_EVENT_CALL_RELEASE_ALL_CNF
Common, 154
TAPI_EVENT_CALL_RELEASE_CNF
Common, 154
TAPI_EVENT_CALL_RETRIEVE_CNF
Common, 154
TAPI_EVENT_CALL_RETRIEVE_IND
Common, 154
TAPI_EVENT_CALL_SEND_DTMF_CNF
Common, 154
TAPI_EVENT_CALL_SETUP_CNF
Common, 153
TAPI_EVENT_CALL_SETUPCONFERENCE_CNF
Common, 154
TAPI_EVENT_CALL_SETUPCONFERENCE_IND
Common, 154
TAPI_EVENT_CALL_SPLITCONFERENCE_CNF
Common, 154
TAPI_EVENT_CALL_TRANSFER_CNF
Common, 154
TAPI_EVENT_CALL_TRANSFER_IND
Common, 154
TAPI_EVENT_CALL_USERINFO_IND
Common, 154
TAPI_EVENT_CALL_WAITING_IND
Common, 154
TAPI_EVENT_CLASS_ALL
Common, 153
TAPI_EVENT_CLASS_CALL_DATA
Common, 153
TAPI_EVENT_CLASS_CALL_VOICE
Common, 153
TAPI_EVENT_CLASS_ERR
Common, 153
TAPI_EVENT_CLASS_NETTEXT
Common, 153
TAPI_EVENT_CLASS_NETWORK
Common, 153
TAPI_EVENT_CLASS_SIM
Common, 153
TAPI_EVENT_CLASS_SS
Common, 153
TAPI_EVENT_CLASS_UNKNOWN
UtilityDS, 208
TAPI_EVENT_CLASS_UTIL
Common, 153
TAPI_EVENT_ERR_RESERVED1_IND
Common, 156
TAPI_EVENT_ERR_RESERVED2_IND
Common, 156
TAPI_EVENT_ERR_RESERVED3_IND
Common, 156
TAPI_EVENT_ERR_RESERVED4_IND
Common, 156
TAPI_EVENT_ERR_RESERVED5_IND
Common, 156
TAPI_EVENT_NETTEXT_CB_INCOM_IND
Common, 155
TAPI_EVENT_NETTEXT_DELETE_STATUS_CNF
Common, 154
TAPI_EVENT_NETTEXT_DELIVERY_REPORT_CNF
Common, 154
TAPI_EVENT_NETTEXT_GET_CB_CONFIG_CNF
Common, 154
TAPI_EVENT_NETTEXT_GET_COUNT_CNF
Common, 154
TAPI_EVENT_NETTEXT_GET_PARAM_CNF
Common, 154
TAPI_EVENT_NETTEXT_GET_SMSBEARER_CNF
Common, 154
TAPI_EVENT_NETTEXT_INCOM_IND
Common, 154
TAPI_EVENT_NETTEXT_MEMORY_STATUS_IND
Common, 155
TAPI_EVENT_NETTEXT_READ_SMS_CNF
Common, 154
TAPI_EVENT_NETTEXT_SAVE_STATUS_CNF
Common, 154
TAPI_EVENT_NETTEXT_SENTSTATUS_CNF
Common, 154
TAPI_EVENT_NETTEXT_SET_REQUEST_CNF
Common, 154
TAPI_EVENT_NETWORK_CANCELMANUALSEARCH_CNF
Common, 155
TAPI_EVENT_NETWORK_CANCELMANUALSELECTION_CNF
Common, 155
TAPI_EVENT_NETWORK_CHANGE_IND
Common, 155
TAPI_EVENT_NETWORK_CONFIG_CNF
Common, 155
TAPI_EVENT_NETWORK_GETNETWORKINFO_CNF
Common, 155
TAPI_EVENT_NETWORK_GETSELECTIONMODE_CNF
Common, 155
TAPI_EVENT_NETWORK_RADIO_INFO_IND

Common, 155
TAPI_EVENT_NETWORK_SEARCH_CNF
Common, 155
TAPI_EVENT_NETWORK_SEARCH_IND
Common, 155
TAPI_EVENT_NETWORK_SELECT_CNF
Common, 155
TAPI_EVENT_NETWORK_SETNWBAND_CNF
Common, 155
TAPI_EVENT_NETWORK_SETNWMODE_CNF
Common, 155
TAPI_EVENT_NETWORK_SETNWORDER_CNF
Common, 155
TAPI_EVENT_NETWORK_SETSELECTIONMODE_CNF
Common, 155
TAPI_EVENT_NETWORK_SETSERVICEDOMAIN_CNF
Common, 155
TAPI_EVENT_SIM_CHANGE_PIN_CNF
Common, 156
TAPI_EVENT_SIM_DISABLE_FDN_CNF
Common, 156
TAPI_EVENT_SIM_DISABLE_PIN_CNF
Common, 156
TAPI_EVENT_SIM_ENABLE_FDN_CNF
Common, 156
TAPI_EVENT_SIM_ENABLE_PIN_CNF
Common, 156
TAPI_EVENT_SIM_FDN_STATUS_IND
Common, 156
TAPI_EVENT_SIM_FILE_INFO_CNF
Common, 156
TAPI_EVENT_SIM_PB_ACCESS_READ_CNF
Common, 156
TAPI_EVENT_SIM_PB_DELETE_CNF
Common, 156
TAPI_EVENT_SIM_PB_ENTRY_INFO_CNF
Common, 156
TAPI_EVENT_SIM_PB_STORAGE_COUNT_CNF
Common, 156
TAPI_EVENT_SIM_PB_UPDATE_CNF
Common, 156
TAPI_EVENT_SIM_READ_FILE_CNF
Common, 156
TAPI_EVENT_SIM_STATUS_IND
Common, 156
TAPI_EVENT_SIM_UNBLOCK_PIN_CNF
Common, 156
TAPI_EVENT_SIM_UPDATE_FILE_CNF
Common, 156
TAPI_EVENT_SIM_VERIFY_PIN_CNF
Common, 156
TAPI_EVENT_SS_BARRING_CHANGEPASS_CNF
Common, 155
TAPI_EVENT_SS_BARRING_CNF
Common, 155
TAPI_EVENT_SS_BARRING_QUERYSTATUS_CNF
Common, 155
TAPI_EVENT_SS_FORWARD_CNF
Common, 155
TAPI_EVENT_SS_FORWARD_QUERYSTATUS_CNF
Common, 155
TAPI_EVENT_SS_GET_PASSWORD_IND
Common, 156
TAPI_EVENT_SS_QUERYCAPABILITY_INFO_CNF
Common, 156
TAPI_EVENT_SS_USSD_CNF
Common, 156
TAPI_EVENT_SS_USSD_IND
Common, 156
TAPI_EVENT_SS_USSD_IND_NOTIFY
Common, 156
TAPI_EVENT_SS_USSD_IND_RSP
Common, 156
TAPI_EVENT_SS_USSDCNF_NOTIFY
Common, 156
TAPI_EVENT_SS_USSDCNF_RSP
Common, 156
TAPI_EVENT_SS_VERIFY_PWD_CNF
Common, 156
TAPI_EVENT_SS_WAITING_CNF
Common, 155
TAPI_EVENT_SS_WAITING_QUERYSTATUS_CNF
Common, 155
TAPI_EVENT_TUNNEL_INFO_CNF
Common, 156
TAPI_EVENT_TUNNEL_INFO_IND
Common, 156
TAPI_EVENT_UNKNOWN
UtilityDS, 208
TAPI_NETTEXT_BEARER_CS_ONLY
NetTextDS, 174
TAPI_NETTEXT_BEARER_CS_PREFERRED
NetTextDS, 174
TAPI_NETTEXT_BEARER_PS_ONLY
NetTextDS, 174
TAPI_NETTEXT_BEARER_PS_PREFERRED
NetTextDS, 174
TAPI_NETTEXT_CALL_BARRED
NetTextDS, 176

TAPI_NETTEXT_CB_MSG_CBS
NetTextDS, 175
TAPI_NETTEXT_CB_MSG_CBS41
NetTextDS, 175
TAPI_NETTEXT_CB_MSG_INVALID
NetTextDS, 175
TAPI_NETTEXT_CB_MSG_SCHEDULE
NetTextDS, 175
TAPI_NETTEXT_CB_PAGE_SIZE_MAX
NetTextDS, 173
TAPI_NETTEXT_CONGESTION
NetTextDS, 176
TAPI_NETTEXT_DEST_ADDRESS_FDN_RESTRICTED
NetTextDS, 175
TAPI_NETTEXT_DESTINAITION_OUTOFSERVICE
NetTextDS, 176
TAPI_NETTEXT_DEVICE_FAILURE
NetTextDS, 175
TAPI_NETTEXT_FACILITY_NOT_IMPLEMENTED
NetTextDS, 176
TAPI_NETTEXT_FACILITY_NOT_SUBSCRIBED
NetTextDS, 176
TAPI_NETTEXT_FACILITY_REJECTED
NetTextDS, 176
TAPI_NETTEXT_GSM_SMS_CBMI_LIST_SIZE_MAX
NetTextDS, 173
TAPI_NETTEXT_GSM_SMS_MSG_NUM_MAX
NetTextDS, 173
TAPI_NETTEXT_IE_NOT_IMPLEMENTED
NetTextDS, 176
TAPI_NETTEXT_INTERWORKING
NetTextDS, 176
TAPI_NETTEXT_INVALID_LOCATION
NetTextDS, 175
TAPI_NETTEXT_INVALID_MANDATORY_INFO
NetTextDS, 176
TAPI_NETTEXT_INVALID_MSG
NetTextDS, 176
TAPI_NETTEXT_INVALID_PARAMETER
NetTextDS, 175
TAPI_NETTEXT_INVALID_PARAMETER_FORMAT
NetTextDS, 174
TAPI_NETTEXT_INVALID_REF_VALUE
NetTextDS, 176
TAPI_NETTEXT_MEMORY_CAPACITY_EXCEEDED
NetTextDS, 176
TAPI_NETTEXT_MEMORY_FAILURE
NetTextDS, 174
TAPI_NETTEXT_MESSAGE_NOT_COMPAT_PROTOCOL
NetTextDS, 176
TAPI_NETTEXT_MESSAGE_TRANSFER_REJECTED
NetTextDS, 176
TAPI_NETTEXT_MESSAGE_TYPE_NOT_IMPLEMENTED
NetTextDS, 176
TAPI_NETTEXT_NETWORK_OUTOFORDER
NetTextDS, 176
TAPI_NETTEXT_NO_NETWORK_RESP
NetTextDS, 175
TAPI_NETTEXT_NO_PREFERRED_BEARER
NetTextDS, 174
TAPI_NETTEXT_NO_SIM
NetTextDS, 175
TAPI_NETTEXT_OPERATION_NOT_SUPPORTED
NetTextDS, 174
TAPI_NETTEXT_OPERATOR_DETERMINED_BARRING
NetTextDS, 176
TAPI_NETTEXT_PHONE_FAILURE
NetTextDS, 174
TAPI_NETTEXT_PROTOCOL_ERROR
NetTextDS, 176
TAPI_NETTEXT_RESEND_ALREADY_DONE
NetTextDS, 175
TAPI_NETTEXT_RESOURCES_UNAVAILABLE
NetTextDS, 176
TAPI_NETTEXT_ROUTING_NOT_AVAILABLE
NetTextDS, 175
TAPI_NETTEXT_SCADDRESS_FDN_RESTRICTED
NetTextDS, 175
TAPI_NETTEXT_SCADDRESS_NOT_AVAILABLE
NetTextDS, 175
TAPI_NETTEXT_SENDSMS_SUCCESS
NetTextDS, 175
TAPI_NETTEXT_SERVICE_RESERVED
NetTextDS, 175
TAPI_NETTEXT_SIM_BUSY
NetTextDS, 174
TAPI_NETTEXT_SIM_FAILURE
NetTextDS, 174
TAPI_NETTEXT_SIM_NOT_READY
NetTextDS, 175
TAPI_NETTEXT_SMDATA_SIZE_MAX
NetTextDS, 173

TAPI_NETTEXT_STATUS_DELIVERED
NetTextDS, 175
TAPI_NETTEXT_STATUS_DELIVERY_UNCONFIRMED
NetTextDS, 175
TAPI_NETTEXT_STATUS_MESSAGE_REPLACED
NetTextDS, 175
TAPI_NETTEXT_STATUS_READ
NetTextDS, 175
TAPI_NETTEXT_STATUS_RESERVED
NetTextDS, 175
TAPI_NETTEXT_STATUS_SENT
NetTextDS, 175
TAPI_NETTEXT_STATUS_UNREAD
NetTextDS, 175
TAPI_NETTEXT_STATUS_UNSENT
NetTextDS, 175
TAPI_NETTEXT_SUCCESS
NetTextDS, 174
TAPI_NETTEXT_TEMPORARY_FAILURE
NetTextDS, 176
TAPI_NETTEXT_UNASSIGNED_NUMBER
NetTextDS, 176
TAPI_NETTEXT_UNKNOWN
NetTextDS, 174
TAPI_NETTEXT_UNKNOWN_SUBSCRIBER
NetTextDS, 176
TAPI_NETTEXT_UNSPECIFIED_SUBSCRIBER
NetTextDS, 176
TAPI_NETWORK_BAND_MODE_ONLY
NetworkDS, 179
TAPI_NETWORK_BAND_MODE_PREFERRED
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_ANY
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_GSM
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_GSM1900
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_GSM850
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_GSM_900_1800
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_WCDMA
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_WCDMA1900
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_WCDMA2100
NetworkDS, 179
TAPI_NETWORK_BAND_TYPE_WCDMA850
NetworkDS, 179
TAPI_NETWORK_CONFIG_SETTINGS_FAILURE_ERR
NetworkDS, 180
TAPI_NETWORK_DLS_ADJ_NO_ADJUSTMENT
NetworkDS, 179
TAPI_NETWORK_DLS_ADJ_PLUS_ONE_HOUR
NetworkDS, 179
TAPI_NETWORK_DLS_PLUS_TWO_HOUR
NetworkDS, 179
TAPI_NETWORK_INTERNAL_FAILURE_ERR
NetworkDS, 180
TAPI_NETWORK_LIST_MAX
NetworkDS, 178
TAPI_NETWORK_MODE_AUTOMATIC
NetworkDS, 179
TAPI_NETWORK_MODE_CDMA
NetworkDS, 179
TAPI_NETWORK_MODE_EMERGENCY
NetworkDS, 179
TAPI_NETWORK_MODE_GSM
NetworkDS, 179
TAPI_NETWORK_MODE_WCDMA
NetworkDS, 179
TAPI_NETWORK_MODE_WLAN
NetworkDS, 179
TAPI_NETWORK_NAME_LEN_MAX
NetworkDS, 178
TAPI_NETWORK_NO_ERR
NetworkDS, 180
TAPI_NETWORK_OFFLINE_ERR
NetworkDS, 180
TAPI_NETWORK_OPERATION_FAILED
NetworkDS, 180
TAPI_NETWORK_ORDER_MAX
NetworkDS, 178
TAPI_NETWORK_PHONE_IN_USE_ERR
NetworkDS, 180
TAPI_NETWORK_SELECTIONMODE_AUTOMATIC
NetworkDS, 180
TAPI_NETWORK_SELECTIONMODE_MANUAL
NetworkDS, 180
TAPI_NETWORK_SERVICE_DOMAIN_AUTOMATIC
NetworkDS, 180
TAPI_NETWORK_SERVICE_DOMAIN_COMBINED
NetworkDS, 180
TAPI_NETWORK_SERVICE_DOMAIN_CS
NetworkDS, 180
TAPI_NETWORK_SERVICE_DOMAIN_PS

NetworkDS, 180
TAPI_NETWORK_SERVICE_LEVEL_EMERGENCY
NetworkDS, 180
TAPI_NETWORK_SERVICE_LEVEL_FULL
NetworkDS, 181
TAPI_NETWORK_SERVICE_LEVEL_NO
NetworkDS, 180
TAPI_NETWORK_SERVICE_LEVEL_SEARCH
NetworkDS, 181
TAPI_NETWORK_SERVICE_TYPE_COMBINED
NetworkDS, 181
TAPI_NETWORK_SERVICE_TYPE_CS
NetworkDS, 181
TAPI_NETWORK_SERVICE_TYPE_LIMITED
NetworkDS, 181
TAPI_NETWORK_SERVICE_TYPE_NO_SVE
NetworkDS, 181
TAPI_NETWORK_SERVICE_TYPE_PS
NetworkDS, 181
TAPI_NETWORK_SPN_LEN_MAX
NetworkDS, 178
TAPI_NETWORK_SYSID_FORMAT_PLMN
NetworkDS, 181
TAPI_NETWORK_SYSID_FORMAT_SID
NetworkDS, 181
TAPI_NETWORK_SYSTEM_CDMA
NetworkDS, 181
TAPI_NETWORK_SYSTEM_EGPRS
NetworkDS, 181
TAPI_NETWORK_SYSTEM_GPRS
NetworkDS, 181
TAPI_NETWORK_SYSTEM_GSM
NetworkDS, 181
TAPI_NETWORK_SYSTEM_GSM_AND_UMTS
NetworkDS, 181
TAPI_NETWORK_SYSTEM_NO_SRV
NetworkDS, 181
TAPI_NETWORK_SYSTEM_PCS1900
NetworkDS, 181
TAPI_NETWORK_SYSTEM_TDMA
NetworkDS, 181
TAPI_NETWORK_SYSTEM_UMTS
NetworkDS, 181
TAPI_REGISTRATION_FAILURE
UtilityDS, 208
TAPI_SIM_ACCESS_ACCESS_CONDITION_NOT_SATISFIED
SIMDS, 189
TAPI_SIM_ACCESS_CARD_ERROR
SIMDS, 189
TAPI_SIM_ACCESS_FAILED
SIMDS, 189
TAPI_SIM_ACCESS_FILE_NOT_FOUND
SIMDS, 189
TAPI_SIM_ACCESS_SUCCESS
SIMDS, 189
TAPI_SIM_ACL_ADM
SIMDS, 189
TAPI_SIM_ACL_ALW
SIMDS, 189
TAPI_SIM_ACL_MAX_LEN
SIMDS, 187
TAPI_SIM_ACL_NEVER
SIMDS, 189
TAPI_SIM_ACL_PIN1
SIMDS, 189
TAPI_SIM_ACL_PIN2
SIMDS, 189
TAPI_SIM_ACL_RFU
SIMDS, 189
TAPI_SIM_BUSY_STATUS
SIMDS, 196
TAPI_SIM_CARD_ERROR_STATUS
SIMDS, 196
TAPI_SIM_CARD_TYPE_GSM
SIMDS, 190
TAPI_SIM_CARD_TYPE_UNKNOWN
SIMDS, 190
TAPI_SIM_CARD_TYPE_USIM
SIMDS, 190
TAPI_SIM_ECC_ALPHA_STRING_LEN_MAX
SIMDS, 187
TAPI_SIM_ECC_CODE_LEN
SIMDS, 187
TAPI_SIM_ECC_ESC_AMBULANCE
SIMDS, 190
TAPI_SIM_ECC_ESC_FIREBRIGADE
SIMDS, 190
TAPI_SIM_ECC_ESC_MARAINEGUARD
SIMDS, 190
TAPI_SIM_ECC_ESC_MOUTAINRESCUE
SIMDS, 190
TAPI_SIM_ECC_ESC_POLICE
SIMDS, 190
TAPI_SIM_ECC_ESC_SPARE
SIMDS, 190
TAPI_SIM_ECC_MAX_LEN
SIMDS, 187
TAPI_SIM_EFILE_ACM
SIMDS, 191
TAPI_SIM_EFILE_ACM_MAX
SIMDS, 191
TAPI_SIM_EFILE_ADN
SIMDS, 190
TAPI_SIM_EFILE_BDN
SIMDS, 191

TAPI_SIM_EFILE_CBMI
    SIMDS, 191
TAPI_SIM_EFILE_CCP
    SIMDS, 191
TAPI_SIM_EFILE_DIR
    SIMDS, 190
TAPI_SIM_EFILE_ECC
    SIMDS, 191
TAPI_SIM_EFILE_EST
    SIMDS, 191
TAPI_SIM_EFILE_EXT1
    SIMDS, 191
TAPI_SIM_EFILE_EXT2
    SIMDS, 191
TAPI_SIM_EFILE_EXT3
    SIMDS, 191
TAPI_SIM_EFILE_EXT4
    SIMDS, 191
TAPI_SIM_EFILE_FDN
    SIMDS, 190
TAPI_SIM_EFILE_GID1
    SIMDS, 191
TAPI_SIM_EFILE_GID2
    SIMDS, 191
TAPI_SIM_EFILE_ICCID
    SIMDS, 190
TAPI_SIM_EFILE_IMG
    SIMDS, 191
TAPI_SIM_EFILE_IMSI
    SIMDS, 191
TAPI_SIM_EFILE_INVALID
    SIMDS, 193
TAPI_SIM_EFILE_LP
    SIMDS, 191
TAPI_SIM_EFILE_MBDN
    SIMDS, 191
TAPI_SIM_EFILE_MSISDN
    SIMDS, 191
TAPI_SIM_EFILE_OTHERS
    SIMDS, 193
TAPI_SIM_EFILE_PHASE
    SIMDS, 191
TAPI_SIM_EFILE_PNN
    SIMDS, 191
TAPI_SIM_EFILE_PUCT
    SIMDS, 191
TAPI_SIM_EFILE_SDN
    SIMDS, 191
TAPI_SIM_EFILE_SMS
    SIMDS, 191
TAPI_SIM_EFILE_SMSP
    SIMDS, 191
TAPI_SIM_EFILE_SMSS
    SIMDS, 191
TAPI_SIM_EFILE_SPN
    SIMDS, 191
TAPI_SIM_EFILE_SST
    SIMDS, 191
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_AAS
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ADN1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANR1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRA1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRB1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_ANRC
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CC
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_CCP1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EMAIL1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_EXT1
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GAS
    SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP

SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_GRP1
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_IAP1
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBC1
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR1
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PBR2
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PSC
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_PUID
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_SNE1
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID
SIMDS, 193
TAPI_SIM_EFILE_TELECOM_PHONEBOOK_UID1
SIMDS, 193
TAPI_SIM_EFILE_USIM_ACM
SIMDS, 191
TAPI_SIM_EFILE_USIM_ACM_MAX
SIMDS, 191
TAPI_SIM_EFILE_USIM_BDN
SIMDS, 192
TAPI_SIM_EFILE_USIM_CBMI
SIMDS, 191
TAPI_SIM_EFILE_USIM_CCP2
SIMDS, 192
TAPI_SIM_EFILE_USIM_ECC
SIMDS, 191
TAPI_SIM_EFILE_USIM_EXT1
SIMDS, 192
TAPI_SIM_EFILE_USIM_EXT2
SIMDS, 192
TAPI_SIM_EFILE_USIM_EXT3
SIMDS, 192
TAPI_SIM_EFILE_USIM_EXT4
SIMDS, 192
TAPI_SIM_EFILE_USIM_EXT5
SIMDS, 192
TAPI_SIM_EFILE_USIM_FDN
SIMDS, 191
TAPI_SIM_EFILE_USIM_HIDDEN_KEY
SIMDS, 192
TAPI_SIM_EFILE_USIM_IMSI
SIMDS, 191
TAPI_SIM_EFILE_USIM_LI
SIMDS, 191
TAPI_SIM_EFILE_USIM_MBDN
SIMDS, 192
TAPI_SIM_EFILE_USIM_MBI
SIMDS, 192
TAPI_SIM_EFILE_USIM_MSISDN
SIMDS, 191
TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_AAS1
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ADN1
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANR1
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRA1
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRB1
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_ANRC1
SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_CC
SIMDS, 192

TAPI_SIM_EFILE_USIM_PHONEBOOK_CCP1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_EMAIL1
 SIMDS, 193
TAPI_SIM_EFILE_USIM_PHONEBOOK_EXT1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GAS1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_GRP1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_IAP1
 SIMDS, 193
TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PBC1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PBR
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PSC
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_PUID
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_SNE1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_UID
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PHONEBOOK_UID1
 SIMDS, 192
TAPI_SIM_EFILE_USIM_PUCT
 SIMDS, 191
TAPI_SIM_EFILE_USIM_SDN
 SIMDS, 192
TAPI_SIM_EFILE_USIM_SMS
 SIMDS, 191
TAPI_SIM_EFILE_USIM_SMSP
 SIMDS, 191
TAPI_SIM_EFILE_USIM_SMSS
 SIMDS, 191
TAPI_SIM_EFILE_USIM_SPN
 SIMDS, 191
TAPI_SIM_EFILE_USIM_UST
 SIMDS, 191
TAPI_SIM_FTYPE_CYCLIC
 SIMDS, 194
TAPI_SIM_FTYPE_DEDICATED
 SIMDS, 194
TAPI_SIM_FTYPE_INVALID_TYPE
 SIMDS, 194
TAPI_SIM_FTYPE_LINEAR_FIXED
 SIMDS, 194
TAPI_SIM_FTYPE_TRANSPARENT
 SIMDS, 194
TAPI_SIM_GET_RESPONSE
 SIMDS, 189
TAPI_SIM_GROUP_IDENTIFIER_LEN_MAX
 SIMDS, 187
TAPI_SIM_GSM_SST_SERVICE_MAX_COUNT
 SIMDS, 187
TAPI_SIM_ICCID_MAX_LEN
 SIMDS, 187
TAPI_SIM_ICCID_PACKET_LEN
 SIMDS, 187
TAPI_SIM_ICCIDLENGTH
 SIMDS, 187
TAPI_SIM_IMSI_LEN
 SIMDS, 187
TAPI_SIM_IMSI_PACKET_LEN
 SIMDS, 187
TAPI_SIM_INCOMPATIBLE_PIN_OPERATION_STATUS
 SIMDS, 196
TAPI_SIM_LI_DANISH
 SIMDS, 194
TAPI_SIM_LI_DUTCH
 SIMDS, 194
TAPI_SIM_LI_ENGLISH
 SIMDS, 194
TAPI_SIM_LI_FINNISH
 SIMDS, 194
TAPI_SIM_LI_FRENCH
 SIMDS, 194
TAPI_SIM_LI_GERMAN
 SIMDS, 194
TAPI_SIM_LI_GREEK
 SIMDS, 194
TAPI_SIM_LI_HUNGARIAN
 SIMDS, 194
TAPI_SIM_LI_ITALIAN
 SIMDS, 194
TAPI_SIM_LI_LANG_UNSPECIFIED
 SIMDS, 194
TAPI_SIM_LI_MAX_LEN
 SIMDS, 187
TAPI_SIM_LI_NORWEGIAN
 SIMDS, 194
TAPI_SIM_LI_PACKET_MAX_LEN

SIMDS, 188
TAPI_SIM_LI_POLISH
SIMDS, 194
TAPI_SIM_LI_PORTUGUESE
SIMDS, 194
TAPI_SIM_LI_SPANISH
SIMDS, 194
TAPI_SIM_LI_SWEDISH
SIMDS, 194
TAPI_SIM_LI_TURKISH
SIMDS, 194
TAPI_SIM_LP_MAX_LEN
SIMDS, 188
TAPI_SIM_LP_PACKET_MAX_LEN
SIMDS, 188
TAPI_SIM_MCC_CODE_LEN
SIMDS, 188
TAPI_SIM_MNC_CODE_LEN
SIMDS, 188
TAPI_SIM_MSIN_CODE_LEN
SIMDS, 188
TAPI_SIM_NPI_DATA_NUMBERING_PLAN
SIMDS, 194
TAPI_SIM_NPI_ERMES_NUMBERING_PLAN
SIMDS, 195
TAPI_SIM_NPI_ISDN_TEL
SIMDS, 194
TAPI_SIM_NPI_NATIONAL
SIMDS, 195
TAPI_SIM_NPI_PRIVATE
SIMDS, 195
TAPI_SIM_NPI_RESERVED_FOR_EXT
SIMDS, 195
TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN
SIMDS, 194
TAPI_SIM_NPI_SVC_CNTR_SPECIFIC_PLAN2
SIMDS, 194
TAPI_SIM_NPI_TELEX
SIMDS, 194
TAPI_SIM_NPI_UNKNOWN
SIMDS, 194
TAPI_SIM_NUM_TEXT_MAX_LEN
SIMDS, 188
TAPI_SIM_PB_ACCESS_CONDITION_NOT_SATISFIED
SIMDS, 195
TAPI_SIM_PB_ACCESS_MODE_ADD
SIMDS, 195
TAPI_SIM_PB_ACCESS_MODE_DEL
SIMDS, 195
TAPI_SIM_PB_ACCESS_MODE_EDIT
SIMDS, 195
TAPI_SIM_PB_ADN
SIMDS, 195
TAPI_SIM_PB_FAIL
SIMDS, 195
TAPI_SIM_PB_FDN
SIMDS, 195
TAPI_SIM_PB_INVALID_INDEX
SIMDS, 195
TAPI_SIM_PB_INVALID_NAME_LENGTH
SIMDS, 195
TAPI_SIM_PB_INVALID_NUMBER_LENGTH
SIMDS, 195
TAPI_SIM_PB_LDN
SIMDS, 195
TAPI_SIM_PB_MSISDN
SIMDS, 195
TAPI_SIM_PB_SDN
SIMDS, 195
TAPI_SIM_PB_SUCCESS
SIMDS, 195
TAPI_SIM_PIN2_INCORRECT_PASSWORD_STATUS
SIMDS, 196
TAPI_SIM_PIN2_REQD_STATUS
SIMDS, 196
TAPI_SIM_PIN_INCORRECT_PASSWORD_STATUS
SIMDS, 196
TAPI_SIM_PIN_LENGTH
SIMDS, 188
TAPI_SIM_PIN_OPERATION_SUCCESS
SIMDS, 196
TAPI_SIM_PIN_STATUS_BLOCKED
SIMDS, 196
TAPI_SIM_PIN_STATUS_DISABLED
SIMDS, 196
TAPI_SIM_PIN_STATUS_ENABLED
SIMDS, 196
TAPI_SIM_PIN_STATUS_PUK_BLOCKED
SIMDS, 196
TAPI_SIM_PIN_STATUS_UNKNOWN
SIMDS, 196
TAPI_SIM_PTYPE_ADM
SIMDS, 196
TAPI_SIM_PTYPE_PIN1
SIMDS, 196
TAPI_SIM_PTYPE_PIN2
SIMDS, 196
TAPI_SIM_PTYPE_PUK1
SIMDS, 196
TAPI_SIM_PTYPE_PUK2
SIMDS, 196
TAPI_SIM_PTYPE_UPIN
SIMDS, 196
TAPI_SIM_PUK2_REQD_STATUS
SIMDS, 196

TAPI_SIM_PUK_INCORRECT_PASSWORD_-STATUS
SIMDS, 196
TAPI_SIM_PUK_REQD_STATUS
SIMDS, 196
TAPI_SIM_READ_BINARY
SIMDS, 189
TAPI_SIM_READ_RECORD
SIMDS, 189
TAPI_SIM_RECORD_ABSOLUTE
SIMDS, 197
TAPI_SIM_RECORD_CURRENT
SIMDS, 197
TAPI_SIM_RECORD_FROM_THE_FIRST
SIMDS, 197
TAPI_SIM_RECORD_NEXT
SIMDS, 197
TAPI_SIM_RECORD_NONE
SIMDS, 197
TAPI_SIM_RECORD_PREVIOUS
SIMDS, 197
TAPI_SIM_RECORD_TO_THE_LAST
SIMDS, 197
TAPI_SIM_SELECT
SIMDS, 189
TAPI_SIM_SMSP_ADDRESS_LEN
NetTextDS, 174
TAPI_SIM_SMSP_ALPHA_ID_LEN_MAX
NetTextDS, 174
TAPI_SIM_SST_MAX_LEN
SIMDS, 188
TAPI_SIM_STATUS
SIMDS, 189
TAPI_SIM_STATUS_CARD_BLOCKED
SIMDS, 190
TAPI_SIM_STATUS_CARD_ERROR
SIMDS, 190
TAPI_SIM_STATUS_CARD_NOT_PRESENT
SIMDS, 190
TAPI_SIM_STATUS_SIM_INIT_COMPLETED
SIMDS, 190
TAPI_SIM_STATUS_SIM_INITIALIZING
SIMDS, 190
TAPI_SIM_STATUS_SIM_PIN_REQUIRED
SIMDS, 190
TAPI_SIM_STATUS_SIM_PUK_REQUIRED
SIMDS, 190
TAPI_SIM_TEXT_ENC_ASCII
SIMDS, 197
TAPI_SIM_TEXT_ENC_GSM7BIT
SIMDS, 197
TAPI_SIM_TEXT_ENC_HEX
SIMDS, 197
TAPI_SIM_TEXT_ENC_UCS2
SIMDS, 197
TAPI_SIM_TON_ABBREVIATED_NUMBER
SIMDS, 197
TAPI_SIM_TON_ALPHA_NUMERIC
SIMDS, 197
TAPI_SIM_TON_DEDICATED_ACCESS
SIMDS, 197
TAPI_SIM_TON_INTERNATIONAL
SIMDS, 197
TAPI_SIM_TON_NATIONAL
SIMDS, 197
TAPI_SIM_TON_NETWORK_SPECIFIC
SIMDS, 197
TAPI_SIM_TON_RESERVED_FOR_EXT
SIMDS, 197
TAPI_SIM_TON_UNKNOWN
SIMDS, 197
TAPI_SIM_UECC_RECORD_COUNT_MAX
SIMDS, 188
TAPI_SIM_UECC_RECORD_LEN_MAX
SIMDS, 188
TAPI_SIM_UPDATE_BINARY
SIMDS, 189
TAPI_SIM_UPDATE_RECORD
SIMDS, 189
TAPI_SS_BAR_GET_PWD
SSDS, 201
TAPI_SS_BAR_NEW_PWD
SSDS, 201
TAPI_SS_BAR_VERIFY_NEW_PWD
SSDS, 201
TAPI_SS_BS_ALL_ASYNCHRONOUS
SSDS, 205
TAPI_SS_BS_ALL_BEARER_SERVICES
SSDS, 205
TAPI_SS_BS_ALL_DATA_CDS
SSDS, 205
TAPI_SS_BS_ALL_DATA_PADCA
SSDS, 205
TAPI_SS_BS_ALL_PLMN_SPEC_BS
SSDS, 205
TAPI_SS_BS_ALL_SYNCHRONOUS
SSDS, 205
TAPI_SS_BS_DATA_ALL_PDS
SSDS, 205
TAPI_SS_BS_DATA_CIRCUIT_ASYNC
SSDS, 205
TAPI_SS_BS_DATA_CIRCUIT_SYNC
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_1
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_2
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_3

SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_4
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_5
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_6
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_7
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_8
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_9
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_A
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_B
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_C
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_D
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_E
SSDS, 205
TAPI_SS_BS_PLMN_SPEC_TELE_F
SSDS, 205
TAPI_SS_CALL_BAR_ACTIVATE
SSDS, 201
TAPI_SS_CALL_BAR_DEACTIVATE
SSDS, 201
TAPI_SS_CALLBARRED
SSDS, 202
TAPI_SS_FACILITYNOTSUPPORTED
SSDS, 202
TAPI_SS_FORWARD_WHEN_ALL_FORWARDING_EV
SSDS, 203
TAPI_SS_FORWARD_WHEN_BUSY_EV
SSDS, 203
TAPI_SS_FORWARD_WHEN_NO_ANSWER_EV
SSDS, 203
TAPI_SS_FORWARD_WHEN_NOT_REACHABLE_EV
SSDS, 203
TAPI_SS_FORWARD_WHEN_UNCONDITIONAL_EV
SSDS, 203
TAPI_SS_GSM_BARR_PW_LEN_MAX
SSDS, 200
TAPI_SS_ILLEGALSSOPERATION_ERROR
SSDS, 202
TAPI_SS_INVALIDDEFLECTEDTONUMBER
SSDS, 202
TAPI_SS_MAXNOMPTYEXCEEDED
SSDS, 202
TAPI_SS_NEGATIVEPWCHECK
SSDS, 202
TAPI_SS_NOTAVAILABLE
SSDS, 202
TAPI_SS_NUMBEROFPWATTEMPTSVIOLATION
SSDS, 202
TAPI_SS_OEM_NOT_SUPPORTED
SSDS, 202
TAPI_SS_PWREGISTRATIONFAILURE
SSDS, 202
TAPI_SS_REJECTEDBYNETWORK
SSDS, 202
TAPI_SS_REJECTEDBYUSER
SSDS, 202
TAPI_SS_RESOURCESNOTAVAILABLE
SSDS, 202
TAPI_SS_SERVICENOTPROVISIONED
SSDS, 202
TAPI_SS_STATUS_ACTIVE
SSDS, 204
TAPI_SS_STATUS_NOTHING
SSDS, 204
TAPI_SS_STATUS_PROVISIONED
SSDS, 204
TAPI_SS_STATUS_QUIESCENT
SSDS, 204
TAPI_SS_STATUS_REGISTERED
SSDS, 204
TAPI_SS_SUBSCRIPTIONVIOLATION
SSDS, 202
TAPI_SS_SUCCESS
SSDS, 202
TAPI_SS_TIMEREXPIRE
SSDS, 202
TAPI_SS_TS_ALL_DATA
SSDS, 205
TAPI_SS_TS_ALL_DATA_TELESERVICES
SSDS, 204
TAPI_SS_TS_ALL_PLMN_SPEC_TELE
SSDS, 204
TAPI_SS_TS_ALL_SPEECH
SSDS, 204
TAPI_SS_TS_ALL_TELE_AND_BEARER_SERVICES
SSDS, 204
TAPI_SS_TS_ALL_TELESERVICES
SSDS, 204
TAPI_SS_TS_ALL_TELESERVICES_EXCEPT_SMS
SSDS, 204
TAPI_SS_TS_DATA_CIRCUIT_ASYNC
SSDS, 205
TAPI_SS_TS_DATA_CIRCUIT_SYNC

SSDS, 205
TAPI_SS_TS_DATA_PACKET
SSDS, 205
TAPI_SS_TS_FACS
SSDS, 204
TAPI_SS_TS_MULTIPLE
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_1
SSDS, 204
TAPI_SS_TS_PLMN_SPEC_TELE_2
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_3
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_4
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_5
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_6
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_7
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_8
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_9
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_A
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_B
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_C
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_D
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_E
SSDS, 205
TAPI_SS_TS_PLMN_SPEC_TELE_F
SSDS, 205
TAPI_SS_TS_TELEPHONY
SSDS, 205
TAPI_SS_TYPE_BARRING
SSDS, 206
TAPI_SS_TYPE_CNAP
SSDS, 206
TAPI_SS_TYPE_FORWARDING
SSDS, 206
TAPI_SS_TYPE_IDENTIFICATION
SSDS, 206
TAPI_SS_TYPE_WAITING
SSDS, 206
TAPI_SS_UNEXPECTEDDATAVALUE
SSDS, 202
TAPI_SS_UNKNOWNALPHABET
SSDS, 202
TAPI_SS_UNKNOWNERROR
SSDS, 202
TAPI_SS_UNKNOWNSUBSCRIBER
SSDS, 202
TAPI_SS_USSD_DATA_SIZE_MAX
SSDS, 200
TAPI_SS_USSDBUSY
SSDS, 202
TAPI_TAPI_SIM_PBOOK_ENTRY_COUNT_MAX
SIMDS, 189
TAPI_USSD_IND_ACTION_REQUIRE
SSDS, 206
TAPI_USSD_IND_NO_ACTION_REQUIRE
SSDS, 206
TAPI_USSD_IND_NOT_SUPPORT
SSDS, 206
TAPI_USSD_IND_NOTIFY
SSDS, 206
TAPI_USSD_IND_OTHER_CLIENT
SSDS, 206
TAPI_USSD_IND_TERMINATED_BY_NET
SSDS, 206
TAPI_USSD_IND_TIME_OUT
SSDS, 206
TAPI_USSD_RES_DISPL_BUSY
SSDS, 206
TAPI_USSD_RES_NONE
SSDS, 206
TAPI_USSD_RES_OK
SSDS, 206
TAPI_USSD_RES_UNKNOWN_ALPHABET
SSDS, 206
TapiCommon.h, 309
TapiResult_t
UtilityDS, 207
TelAppCallback
Common, 152
TelCall.h, 312
TelCallActiveState_t
CallDS, 167
TelCallAlertingPattern_t
CallDS, 167
TelCallAnswerType_t
CallDS, 167
TelCallCause_t
CallDS, 168
TelCallCugInfo_t, 209
TelCallCugInfo_t
bCugFlag, 209
Index, 209
Option, 209
TelCallCugOptions_t
CallDS, 170
TelCallEndInfo_t, 210

TelCallEndInfo_t
  CallEndCause, 210
  CallEndTime, 210
  CallStartTime, 210
  pCallHandle, 210
TelCallIdentityMode_t
  CallDS, 170
TelCallIncomingCallInfo_t, 211
TelCallIncomingCallInfo_t
  AlertingPattern, 212
  CallHandle, 211
  CallingNameInfo, 211
  CallType, 211
  CliPresentationIndicator, 211
  RedirectInfo, 212
  Signal, 212
  szCallingPartyNumber, 211
  szCallingPartySubAddress, 211
  UusInfo, 212
TelCallingNameInfo_t, 213
TelCallingNameInfo_t
  NameMode, 213
  szNameData, 213
TelCallNameMode_t
  CallDS, 170
TelCallRedirectionInfo_t, 214
TelCallRedirectionInfo_t
  szRedirectedNumber, 214
  szRedirectSubAddress, 214
TelCallSetupParams_t, 215
TelCallSetupParams_t
  CallType, 215
  IdentityMode, 215
  pCugInfo, 215
  pUusInfo, 215
  szCalledPartySubAddress, 215
  szNumber, 215
TelCallStatus_t, 217
TelCallStatus_t
  bConferenceState, 217
  bMoCall, 217
  CallState, 217
  CallType, 217
  pNumber, 217
TelCallType_t
  CallDS, 171
TelCallUserDataInfo_t, 218
TelCallUserDataInfo_t
  CallHandle, 218
  UusInfo, 218
TelCallUusInfo_t, 219
TelCallUusInfo_t
  bMoreDataPresent, 219
  ProtocolType, 219
  UusData, 219
  UusDataLen, 219
  UusType, 219
TelCallUusProtocolType_t
  CallDS, 171
TelCallUusType_t
  CallDS, 171
TeleCommService
  TelSsCallBarringRecord_t, 276
  TelSsCallWaitingRecord_t, 277
  TelSsForwardRecord_t, 281
TelNetText.h, 315
TelNetwork.h, 317
TelNetworkBand_t
  NetworkDS, 178
TelNetworkBandPreferred_t
  NetworkDS, 179
TelNetworkConfigSettings_t, 220
TelNetworkConfigSettings_t
  bAutoSelection, 220
  bPowerOnGprsAttach, 220
  NetworkMode, 220
  NetworkOrder, 220
  NoOfNwModePresent, 220
  ServiceDomain, 220
TelNetworkDayLightSavingAdj_t
  NetworkDS, 179
TelNetworkInfo_t, 221
TelNetworkInfo_t
  bDisplayRplmn, 221
  bDisplaySpn, 221
  bRoaming, 221
  ServiceLevel, 221
  SysId, 221
TelNetworkList_t, 222
TelNetworkList_t
  NumOfNwAvailable, 222
  SystemList, 222
TelNetworkMode_t
  NetworkDS, 179
TelNetworkOperationCause_t
  NetworkDS, 179
TelNetworkRadioInfo_t, 223
TelNetworkRadioInfo_t
  RssiBar, 223
  RxLevel, 223
  RxQuality, 223
TelNetworkSelectionMode_t
  NetworkDS, 180
TelNetworkServiceDomain_t
  NetworkDS, 180
TelNetworkServiceLevel_t
  NetworkDS, 180
TelNetworkServiceType_t

    NetworkDS, 181
TelNetworkSystemId_t, 224
TelNetworkSystemId_t
    bEquivalentPlmn, 225
    bHome, 225
    bMncIncludesPcsDigit, 225
    bNetworkDlsAvail, 225
    bNetworkIdentityAvail, 225
    bNetworkTimeAvail, 225
    bNetworkTimeZoneAvail, 225
    bPlmnForbidden, 224
    NetworkMode, 224
    NetworkTimeZone, 225
    ServiceType, 224
    SysId, 224
    SysIdFormat, 224
    SystemType, 224
    szSpn, 225
    szSystemName, 225
TelNetworkSystemIdFormat_t
    NetworkDS, 181
TelNetworkSystemType_t
    NetworkDS, 181
TelNetworkTimeAndZone_t, 226
TelNetworkTimeAndZone_t
    Day, 226
    DaylightSave, 227
    Hour, 226
    Minute, 226
    Month, 226
    Second, 226
    TimeZone, 226
    Year, 226
TelSim.h, 319
TelSimAccessResult_t
    SIMDS, 189
TelSimAccessType_t
    SIMDS, 189
TelSimAcl_t
    SIMDS, 189
TelSimApnData_t, 228
TelSimApnData_t
    Acl, 228
    AclLen, 228
TelSimCallbackEvent_t, 229
TelSimCallbackEvent_t
    DataLen, 229
    EventStatus, 229
    EventType, 229
    pData, 229
    RequestId, 229
TelSimCardStatus_t
    SIMDS, 189
TelSimCardType_t
    SIMDS, 190
TelSimDeviceContext_t, 230
TelSimDeviceContext_t
    SimDeviceCtxData, 230
    SimStatusData, 230
TelSimDeviceContextData_t, 231
TelSimDeviceContextData_t
    apnCount, 231
    ApnData, 232
    bAclEnable, 231
    bBdnEnable, 231
    bFdnEnable, 231
    bImsiInvalidated, 231
    bIsDeviceCtxdataValid, 231
    Gid1Info, 232
    Gid2Info, 232
    SstData, 232
    SstInfo, 232
    UstInfo, 232
TelSimEccData_t, 233
TelSimEccData_t
    EccInfo, 233
    UeccInfo, 233
TelSimEccEmergencyServiceInfo_t
    SIMDS, 190
TelSimEccInfo_t, 234
TelSimEccInfo_t
    szEcc1, 234
    szEcc2, 234
    szEcc3, 234
    szEcc4, 234
    szEcc5, 234
TelSimEvent_t, 235
TelSimEvent_t
    DataLen, 235
    EventStatus, 235
    pData, 235
    RequestId, 235
TelSimFileAccessGeneralResp_t, 236
TelSimFileAccessGeneralResp_t
    AccessType, 236
    FileId, 236
TelSimFileID_t, 237
TelSimFileID_t
    FileName, 237
    szPath, 237
TelSimFileInfo_t, 238
TelSimFileInfo_t
    AcInvalidate, 238
    AcRead, 238
    AcRehabilitate, 238
    AcUpdate, 238
    bValid, 238
    FileId, 238

- FileSize, 239
- FileType, 238
- RecordCount, 239
- RecordLen, 239

TelSimFileName_t
- SIMDS, 190

TelSimFileType_t
- SIMDS, 193

TelSimGroupIdentifierInfo_t, 240

TelSimGroupIdentifierInfo_t
- GroupIdentifierLen, 240
- szGroupIdentifier, 240

TelSimIccIdInfo_t, 241

TelSimIccIdInfo_t
- Iccid, 241
- IccidLength, 241

TelSimImsiInfo_t, 242

TelSimImsiInfo_t
- bValid, 242
- szMcc, 242
- szMnc, 242
- szMsin, 242

TelSimLangIndicationCode_t
- SIMDS, 194

TelSimLanguageInfo_t, 243

TelSimLanguageInfo_t
- lp, 243
- LpCount, 243

TelSimNumberingPlanIdentity_t
- SIMDS, 194

TelSimPb2gData_t, 244

TelSimPb2gData_t
- Name, 244
- NameEncryptType, 244
- NameLen, 244
- Number, 244
- NumLen, 244
- NumType, 244

TelSimPbAccessMode_t
- SIMDS, 195

TelSimPbAccessResult_t
- SIMDS, 195

TelSimPbEntryInfo_t, 245

TelSimPbEntryInfo_t
- PbIndexMax, 245
- PbIndexMin, 245
- PbNumLenMax, 245
- PbTextLenMax, 245
- StorageFileType, 245

TelSimPbFileType_t
- SIMDS, 195

TelSimPbReadResp_t, 246

TelSimPbReadResp_t
- PbAccessResult, 246
- PbStorageData, 246

TelSimPbRecordData_t, 247

TelSimPbRecordData_t
- Index, 247
- pPbData, 247
- StorageFileType, 247

TelSimPbStorageInfo_t, 248

TelSimPbStorageInfo_t
- StorageFileType, 248
- TotalRecordCount, 248
- UsedRecordCount, 248

TelSimPbUpdateResp_t, 249

TelSimPbUpdateResp_t
- PbAccessResult, 249
- PbIndex, 249
- PbStorageInfo, 249

TelSimPinData_t, 250

TelSimPinData_t
- PinType, 250
- szPinCode, 250

TelSimPinInfo_t, 251

TelSimPinInfo_t
- PinType, 251
- RemainingAttempts, 251

TelSimPinOperationResult_t
- SIMDS, 195

TelSimPinStatus_t
- SIMDS, 196

TelSimPinType_t
- SIMDS, 196

TelSimReadFileData_t, 252

TelSimReadFileData_t
- Data, 252
- DataLen, 252
- FileId, 252

TelSimReadFileRequestData_t, 253

TelSimReadFileRequestData_t
- DataLength, 254
- FileID, 253
- FileType, 253
- Offset, 253
- RecordIndex, 253
- RecordIndexFrom, 253
- RecordIndexTo, 253
- RecordMode, 253

TelSimRecordMode_t
- SIMDS, 196

TelSimServiceTable_t, 255

TelSimServiceTable_t
- bAdn, 256
- bAoc, 256
- bAutoAnswerForeMLPP, 258
- bBdn, 258
- bCallControl, 258

  - bCcp, 256
  - bCellBroadCastMsgId, 257
  - bCellBroadcastMsgIdRanges, 258
  - bCoOperativeNetworkList, 259
  - bCPBCCHInfo, 260
  - bDePersonalizedControlKeys, 259
  - bEnhancedMultilevelPrecedencePreemptionService, 258
  - bExtenedCapaConfigParams, 260
  - bExtension1, 257
  - bExtension2, 257
  - bExtension3, 257
  - bExtension4, 259
  - bFdn, 256
  - bGPRS, 259
  - bGroupIdLevel1, 257
  - bGroupIdLevel2, 257
  - bHplmnSelectorAccessTech, 260
  - bImage, 259
  - bInvestigationScan, 260
  - blnd, 257
  - bMenuSelection, 258
  - bMexe, 260
  - bMoSMSControl, 259
  - bMsisdn, 256
  - bNetworkAlertIndInMS, 259
  - bOperatorControlledPlmnSelectorAccessTech, 260
  - bPIN1DisableFunction, 256
  - bPlmnSelector, 256
  - bProactiveSimCommand, 258
  - bRfu1, 256
  - bRfu2, 257
  - bRplmnLastUsedAccessTech, 260
  - bRunAtCommand, 259
  - bSdn, 257
  - bSms, 256
  - bSmsCbdDataDownload, 258
  - bSmsParams, 257
  - bSmsPpDataDownload, 258
  - bSmsr, 259
  - bSolsa, 259
  - bSpn, 257
  - bUserControlledPLMNSelectorAccessTech, 260
  - bUSSDSupportedInCallControl, 259
  - bVbsGroupIDList, 258
  - bVgcsGroupIdList, 258
- TelSimSstInfo_t, 261
- TelSimSstInfo_t
  - Sst, 261
  - SstData, 261
  - SstLen, 261
- TelSimStatusData_t, 262
- TelSimStatusData_t
  - CardStatus, 262
  - CardType, 262
  - EccRecord, 262
  - EccRecordCount, 262
  - Iccid, 263
  - IccidLen, 263
  - LpInfo, 263
  - Pin1Status, 262
  - Pin2Status, 262
  - UpinStatus, 262
- TelSimTextEncrypt_t
  - SIMDS, 197
- TelSimTypeOfNum_t
  - SIMDS, 197
- TelSimUpdateFileRequestData_t, 264
- TelSimUpdateFileRequestData_t
  - DataLen, 264
  - FileId, 264
  - FileType, 264
  - Offset, 264
  - RecordIndex, 264
  - RecordMode, 264
  - szData, 264
- TelSimUstInfo_t, 266
- TelSimUstInfo_t
  - UsimSst, 266
  - UsimSstLen, 266
  - UstData, 266
- TelSmsAddressInfo_t, 267
- TelSmsAddressInfo_t
  - DialNumLen, 267
  - Npi, 267
  - szDiallingNum, 267
  - Ton, 267
- TelSmsBearerType_t
  - NetTextDS, 174
- TelSmsCause_t
  - NetTextDS, 174
- TelSmsCbConfig_t, 268
- TelSmsCbConfig_t
  - bCBEnabled, 268
  - MsgIdCount, 268
  - MsgIDs, 268
  - SelectedId, 268
- TelSmsCbMsg_t, 269
- TelSmsCbMsg_t
  - CbMsgType, 269
  - Length, 269
  - szMsgData, 269
- TelSmsCbMsgType_t
  - NetTextDS, 174
- TelSmsData_t, 270
- TelSmsData_t

- MsgStatus, 270
- SimIndex, 270
- SmsData, 270

TelSmsDatapackageInfo_t, 271

TelSmsDatapackageInfo_t
- MsgLength, 271
- pSCA, 271
- szData, 271

TelSmsMsgStatus_t
- NetTextDS, 175

TelSmsParams_t, 272

TelSmsParams_t
- AlphaIdLen, 272
- ParamIndicator, 272
- RecordIndex, 272
- RecordLen, 272
- szAlphaId, 272
- TpDataCodingScheme, 273
- TpDestAddr, 272
- TpProtocolId, 273
- TpSvcCntrAddr, 273
- TpValidityPeriod, 273

TelSmsResponse_t
- NetTextDS, 175

TelSmsStoredMsgCountInfo_t, 274

TelSmsStoredMsgCountInfo_t
- IndexList, 274
- TotalCount, 274
- UsedCount, 274

TelSs.h, 324

TelSsBarPwdType_t
- SSDS, 201

TelSsCallBarOperationMode_t
- SSDS, 201

TelSsCallBarringInfo_t, 275

TelSsCallBarringInfo_t
- CallType, 275
- Mode, 275
- szPassword, 275
- Type, 275

TelSsCallBarringRecord_t, 276

TelSsCallBarringRecord_t
- Flavour, 276
- Status, 276
- TeleCommService, 276

TelSsCallBarType_t
- SSDS, 201

TelSsCallType_t
- SSDS, 201

TelSsCallWaitingRecord_t, 277

TelSsCallWaitingRecord_t
- Status, 277
- TeleCommService, 277

TelSsCapabilityInfo_t, 278

TelSsCapabilityInfo_t
- bCallHoldSupported, 278
- bCBSupported, 278
- bCFSupported, 278
- bCWSupported, 278
- bECTSupported, 278
- bMPTYSupported, 278

TelSsCause_t
- SSDS, 202

TelSsCliRecord_t, 279

TelSsCliRecord_t
- CliStatus, 279
- IdentificationType, 279

TelSsCliStatus_t
- SSDS, 202

TelSsForwardInfo_t, 280

TelSsForwardInfo_t
- Condition, 280
- Mode, 280
- szPhoneNumber, 280
- Type, 280

TelSsForwardMode_t
- SSDS, 203

TelSsForwardRecord_t, 281

TelSsForwardRecord_t
- bCallForwardingNumberPresent, 281
- ForwardCondition, 281
- Status, 281
- szCallForwardingNumber, 281
- TeleCommService, 281

TelSsForwardType_t
- SSDS, 203

TelSsForwardWhen_t
- SSDS, 203

TelSsInfo_t, 282

TelSsInfo_t
- NumberOfRecords, 282
- SsRecord, 282
- SsStatus, 282
- SsTeleserviceType, 282
- SsType, 282

TelSsLineIdentificationType_t
- SSDS, 203

TelSsRecord_t, 283

TelSsRecord_t
- BarringRecord, 283
- CliRecord, 283
- ForwardingRecord, 283
- WaitingRecord, 283

TelSsRestrictIDStatus_t
- SSDS, 204

TelSsStatus_t
- SSDS, 204

TelSsTeleService_t

SSDS, 204
TelSsType_t
    SSDS, 205
TelSsUssdIndType_t
    SSDS, 206
TelSsUssdMsgInfo_t, 284
TelSsUssdMsgInfo_t
    Dcs, 284
    IndType, 284
    szUssdString, 284
    UssdStringLength, 284
TelSsUssdRespType_t
    SSDS, 206
TelTapiCallAnswer, 30
    TelTapiCallAnswer, 30
TelTapiCallAnswer
    TelTapiCallAnswer, 30
TelTapiCallExplicitTransfer, 43
    TelTapiCallExplicitTransfer, 43
TelTapiCallExplicitTransfer
    TelTapiCallExplicitTransfer, 43
TelTapiCallGetConferenceList, 47
    TelTapiCallGetConferenceList, 47
TelTapiCallGetConferenceList
    TelTapiCallGetConferenceList, 47
TelTapiCallGetDuration, 46
    TelTapiCallGetDuration, 46
TelTapiCallGetDuration
    TelTapiCallGetDuration, 46
TelTapiCallGetStatus, 45
    TelTapiCallGetStatus, 45
TelTapiCallGetStatus
    TelTapiCallGetStatus, 45
TelTapiCallHold, 38
    TelTapiCallHold, 38
TelTapiCallHold
    TelTapiCallHold, 38
TelTapiCallJoin, 39
    TelTapiCallJoin, 39
TelTapiCallJoin
    TelTapiCallJoin, 39
TelTapiCallRelease, 32
    TelTapiCallRelease, 32
TelTapiCallRelease
    TelTapiCallRelease, 32
TelTapiCallReleaseAll, 33
    TelTapiCallReleaseAll, 33
TelTapiCallReleaseAll
    TelTapiCallReleaseAll, 33
TelTapiCallRetrieve, 37
    TelTapiCallRetrieve, 37
TelTapiCallRetrieve
    TelTapiCallRetrieve, 37
TelTapiCallSendDTMF, 35
    TelTapiCallSendDTMF, 35
TelTapiCallSendDTMF
    TelTapiCallSendDTMF, 35
TelTapiCallSetup, 28
    TelTapiCallSetup, 28
TelTapiCallSetup
    TelTapiCallSetup, 28
TelTapiCallSplit, 41
    TelTapiCallSplit, 41
TelTapiCallSplit
    TelTapiCallSplit, 41
TelTapiDeRegister, 161
    TelTapiDeRegister, 161
TelTapiDeRegister
    TelTapiDeRegister, 161
TelTapiErrorInfo_t, 285
TelTapiErrorInfo_t
    Length, 285
    pErrorData, 285
TelTapiEvent_t, 286
TelTapiEvent_t
    EventClass, 286
    EventType, 286
    pData, 286
    RequestId, 286
    Status, 286
TelTapiEventClass_t
    Common, 153
TelTapiEventType_t
    Common, 153
TelTapiGetVersionInfo, 164
    TelTapiGetVersionInfo, 164
TelTapiGetVersionInfo
    TelTapiGetVersionInfo, 164
TelTapiNwCancelManualSearch, 74
    TelTapiNwCancelManualSearch, 74
TelTapiNwCancelManualSearch
    TelTapiNwCancelManualSearch, 74
TelTapiNwCancelManualSelection, 75
    TelTapiNwCancelManualSelection, 75
TelTapiNwCancelManualSelection
    TelTapiNwCancelManualSelection, 75
TelTapiNwGetNetworkInfo, 76
    TelTapiNwGetNetworkInfo, 76
TelTapiNwGetNetworkInfo
    TelTapiNwGetNetworkInfo, 76
TelTapiNwGetSelectionMode, 80
    TelTapiNwGetSelectionMode, 80
TelTapiNwGetSelectionMode
    TelTapiNwGetSelectionMode, 80
TelTapiNwSearchNetwork, 77
    TelTapiNwSearchNetwork, 77
TelTapiNwSearchNetwork
    TelTapiNwSearchNetwork, 77

TelTapiNwSelectNetwork, 72
  TelTapiNwSelectNetwork, 72
TelTapiNwSelectNetwork
  TelTapiNwSelectNetwork, 72
TelTapiNwSetConfiguration, 70
  TelTapiNwSetConfiguration, 70
TelTapiNwSetConfiguration
  TelTapiNwSetConfiguration, 70
TelTapiNwSetNetworkBand, 87
  TelTapiNwSetNetworkBand, 87
TelTapiNwSetNetworkBand
  TelTapiNwSetNetworkBand, 87
TelTapiNwSetNetworkMode, 85
  TelTapiNwSetNetworkMode, 85
TelTapiNwSetNetworkMode
  TelTapiNwSetNetworkMode, 85
TelTapiNwSetNetworkOrder, 83
  TelTapiNwSetNetworkOrder, 83
TelTapiNwSetNetworkOrder
  TelTapiNwSetNetworkOrder, 83
TelTapiNwSetSelectionMode, 78
  TelTapiNwSetSelectionMode, 78
TelTapiNwSetSelectionMode
  TelTapiNwSetSelectionMode, 78
TelTapiNwSetServiceDomain, 81
  TelTapiNwSetServiceDomain, 81
TelTapiNwSetServiceDomain
  TelTapiNwSetServiceDomain, 81
TelTapiRegisterClass, 157
  TelTapiRegisterClass, 157
TelTapiRegisterClass
  TelTapiRegisterClass, 157
TelTapiRegisterEvent, 159
  TelTapiRegisterEvent, 159
TelTapiRegisterEvent
  TelTapiRegisterEvent, 159
TelTapiSimChangePIN, 97
  TelTapiSimChangePIN, 97
TelTapiSimChangePIN
  TelTapiSimChangePIN, 97
TelTapiSimDisableFDN, 104
  TelTapiSimDisableFDN, 104
TelTapiSimDisableFDN
  TelTapiSimDisableFDN, 104
TelTapiSimDisablePIN, 99
  TelTapiSimDisablePIN, 99
TelTapiSimDisablePIN
  TelTapiSimDisablePIN, 99
TelTapiSimEnableFDN, 105
  TelTapiSimEnableFDN, 105
TelTapiSimEnableFDN
  TelTapiSimEnableFDN, 105
TelTapiSimEnablePIN, 101
  TelTapiSimEnablePIN, 101
TelTapiSimEnablePIN
  TelTapiSimEnablePIN, 101
TelTapiSimGetCardType, 117
  TelTapiSimGetCardType, 117
TelTapiSimGetCardType
  TelTapiSimGetCardType, 117
TelTapiSimGetDevCtxdata, 106
  TelTapiSimGetDevCtxdata, 106
TelTapiSimGetDevCtxdata
  TelTapiSimGetDevCtxdata, 106
TelTapiSimGetEccInfo, 114
  TelTapiSimGetEccInfo, 114
TelTapiSimGetEccInfo
  TelTapiSimGetEccInfo, 114
TelTapiSimGetFDNStatus, 110
  TelTapiSimGetFDNStatus, 110
TelTapiSimGetFDNStatus
  TelTapiSimGetFDNStatus, 110
TelTapiSimGetFileInfo, 90
  TelTapiSimGetFileInfo, 90
TelTapiSimGetFileInfo
  TelTapiSimGetFileInfo, 90
TelTapiSimGetIccIdInfo, 107
  TelTapiSimGetIccIdInfo, 107
TelTapiSimGetIccIdInfo
  TelTapiSimGetIccIdInfo, 107
TelTapiSimGetImsiInfo, 113
  TelTapiSimGetImsiInfo, 113
TelTapiSimGetImsiInfo
  TelTapiSimGetImsiInfo, 113
TelTapiSimGetLanguageIndInfo, 116
  TelTapiSimGetLanguageIndInfo, 116
TelTapiSimGetLanguageIndInfo
  TelTapiSimGetLanguageIndInfo, 116
TelTapiSimGetLanguagePrefInfo, 115
  TelTapiSimGetLanguagePrefInfo, 115
TelTapiSimGetLanguagePrefInfo
  TelTapiSimGetLanguagePrefInfo, 115
TelTapiSimGetPIN2Status, 109
  TelTapiSimGetPIN2Status, 109
TelTapiSimGetPIN2Status
  TelTapiSimGetPIN2Status, 109
TelTapiSimGetPINStatus, 108
  TelTapiSimGetPINStatus, 108
TelTapiSimGetPINStatus
  TelTapiSimGetPINStatus, 108
TelTapiSimGetSstInfo, 111
  TelTapiSimGetSstInfo, 111
TelTapiSimGetSstInfo
  TelTapiSimGetSstInfo, 111
TelTapiSimGetUstInfo, 112
  TelTapiSimGetUstInfo, 112
TelTapiSimGetUstInfo
  TelTapiSimGetUstInfo, 112

TelTapiSimPBDeleteRecord, 125
    TelTapiSimPBDeleteRecord, 125
TelTapiSimPBDeleteRecord
    TelTapiSimPBDeleteRecord, 125
TelTapiSimPBGetEntryInfo, 120
    TelTapiSimPBGetEntryInfo, 120
TelTapiSimPBGetEntryInfo
    TelTapiSimPBGetEntryInfo, 120
TelTapiSimPBGetStorageCount, 118
    TelTapiSimPBGetStorageCount, 118
TelTapiSimPBGetStorageCount
    TelTapiSimPBGetStorageCount, 118
TelTapiSimPBReadRecord, 122
    TelTapiSimPBReadRecord, 122
TelTapiSimPBReadRecord
    TelTapiSimPBReadRecord, 122
TelTapiSimPBUpdateRecord, 124
    TelTapiSimPBUpdateRecord, 124
TelTapiSimPBUpdateRecord
    TelTapiSimPBUpdateRecord, 124
TelTapiSimReadFile, 92
    TelTapiSimReadFile, 92
TelTapiSimReadFile
    TelTapiSimReadFile, 92
TelTapiSimUnblockPIN, 102
    TelTapiSimUnblockPIN, 102
TelTapiSimUnblockPIN
    TelTapiSimUnblockPIN, 102
TelTapiSimUpdateFile, 94
    TelTapiSimUpdateFile, 94
TelTapiSimUpdateFile
    TelTapiSimUpdateFile, 94
TelTapiSimVerifyPIN, 95
    TelTapiSimVerifyPIN, 95
TelTapiSimVerifyPIN
    TelTapiSimVerifyPIN, 95
TelTapiSmsCBGetConfig, 60
    TelTapiSmsCBGetConfig, 60
TelTapiSmsCBGetConfig
    TelTapiSmsCBGetConfig, 60
TelTapiSmsCBSetConfig, 58
    TelTapiSmsCBSetConfig, 58
TelTapiSmsCBSetConfig
    TelTapiSmsCBSetConfig, 58
TelTapiSmsDelete, 56
    TelTapiSmsDelete, 56
TelTapiSmsDelete
    TelTapiSmsDelete, 56
TelTapiSmsGetCount, 57
    TelTapiSmsGetCount, 57
TelTapiSmsGetCount
    TelTapiSmsGetCount, 57
TelTapiSmsGetParameters, 65
    TelTapiSmsGetParameters, 65
TelTapiSmsGetParameters
    TelTapiSmsGetParameters, 65
TelTapiSmsGetPreferredBearer, 64
    TelTapiSmsGetPreferredBearer, 64
TelTapiSmsGetPreferredBearer
    TelTapiSmsGetPreferredBearer, 64
TelTapiSmsGetSCA, 62
    TelTapiSmsGetSCA, 62
TelTapiSmsGetSCA
    TelTapiSmsGetSCA, 62
TelTapiSmsRead, 52
    TelTapiSmsRead, 52
TelTapiSmsRead
    TelTapiSmsRead, 52
TelTapiSmsSend, 50
    TelTapiSmsSend, 50
TelTapiSmsSend
    TelTapiSmsSend, 50
TelTapiSmsSendDeliveryReport, 68
    TelTapiSmsSendDeliveryReport, 68
TelTapiSmsSendDeliveryReport
    TelTapiSmsSendDeliveryReport, 68
TelTapiSmsSetParameters, 66
    TelTapiSmsSetParameters, 66
TelTapiSmsSetParameters
    TelTapiSmsSetParameters, 66
TelTapiSmsSetPreferredBearer, 63
    TelTapiSmsSetPreferredBearer, 63
TelTapiSmsSetPreferredBearer
    TelTapiSmsSetPreferredBearer, 63
TelTapiSmsSetSCA, 61
    TelTapiSmsSetSCA, 61
TelTapiSmsSetSCA
    TelTapiSmsSetSCA, 61
TelTapiSmsWrite, 54
    TelTapiSmsWrite, 54
TelTapiSmsWrite
    TelTapiSmsWrite, 54
TelTapiSsChangeBarPassword, 132
    TelTapiSsChangeBarPassword, 132
TelTapiSsChangeBarPassword
    TelTapiSsChangeBarPassword, 132
TelTapiSsGetCallBarringStatus, 130
    TelTapiSsGetCallBarringStatus, 130
TelTapiSsGetCallBarringStatus
    TelTapiSsGetCallBarringStatus, 130
TelTapiSsGetCallForwardStatus, 136
    TelTapiSsGetCallForwardStatus, 136
TelTapiSsGetCallForwardStatus
    TelTapiSsGetCallForwardStatus, 136
TelTapiSsGetCallWaitingStatus, 140
    TelTapiSsGetCallWaitingStatus, 140
TelTapiSsGetCallWaitingStatus
    TelTapiSsGetCallWaitingStatus, 140

TelTapiSsGetRestrictCallId, 145
  TelTapiSsGetRestrictCallId, 145
TelTapiSsGetRestrictCallId
  TelTapiSsGetRestrictCallId, 145
TelTapiSsQueryCapabilityInfo, 146
  TelTapiSsQueryCapabilityInfo, 146
TelTapiSsQueryCapabilityInfo
  TelTapiSsQueryCapabilityInfo, 146
TelTapiSsSetCallBarring, 128
  TelTapiSsSetCallBarring, 128
TelTapiSsSetCallBarring
  TelTapiSsSetCallBarring, 128
TelTapiSsSetCallForward, 134
  TelTapiSsSetCallForward, 134
TelTapiSsSetCallForward
  TelTapiSsSetCallForward, 134
TelTapiSsSetCallWaiting, 138
  TelTapiSsSetCallWaiting, 138
TelTapiSsSetCallWaiting
  TelTapiSsSetCallWaiting, 138
TelTapiSsSetRestrictCallId, 144
  TelTapiSsSetRestrictCallId, 144
TelTapiSsSetRestrictCallId
  TelTapiSsSetRestrictCallId, 144
TelTapiSsVerifyBarPwdResponse, 142
  TelTapiSsVerifyBarPwdResponse, 142
TelTapiSsVerifyBarPwdResponse
  TelTapiSsVerifyBarPwdResponse, 142
TelTapiTunnelInfo_t, 287
TelTapiTunnelInfo_t
  pTunnelData, 287
  TunnelDataLen, 287
TelTapiTunnelReq, 163
  TelTapiTunnelReq, 163
TelTapiTunnelReq
  TelTapiTunnelReq, 163
TelTapiUssdSendRequest, 147
  TelTapiUssdSendRequest, 147
TelTapiUssdSendRequest
  TelTapiUssdSendRequest, 147
TelTapiUssdSendUserResponse, 149
  TelTapiUssdSendUserResponse, 149
TelTapiUssdSendUserResponse
  TelTapiUssdSendUserResponse, 149
TelTapiVersionInfo_t, 288
TelTapiVersionInfo_t
  MajorVersion, 288
  MinorVersion, 288
TelUsimEccInfo_t, 289
TelUsimEccInfo_t
  bUsed, 289
  EccEmergencyServiceInfo, 289
  EccLen, 289
  szEcc, 289
  szEccAlphaIdentifier, 289
TelUSimServiceTable_t, 290
TelUSimServiceTable_t
  bAcl, 294
  bAoc, 292
  bAutoAnswerForeMLPP, 293
  bBdn, 291
  bCallForwardingIndicationStatus, 295
  bCcp, 292
  bCellBroadcastMsgId, 292
  bCellBroadcastMsgIdRanges, 292
  bCoOperativeNetworkList, 294
  bCpbcchiInformation, 294
  bDePersonalizedControlKeys, 294
  bEnabledServiceTable, 294
  bEnhancedMultilevelPrecedencePreemptionService, 293
  bExtension2, 291
  bExtension3, 291
  bExtension4, 291
  bExtension5, 295
  bExtension8, 296
  bFdn, 291
  bGroupIdLevel1, 292
  bGroupIdLevel2, 292
  bGsmAccess, 293
  bGsmSecurityContext, 294
  bHplmnSelectorAccessTech, 295
  bIciIct, 292
  bImage, 293
  bInvestigationScan, 294
  bLocalPhoneBook, 291
  bMailBoxDiallingNumbers, 295
  bMexe, 295
  bMms, 296
  bMmsUserConnectivityParameters, 296
  bMsgWaitingIndStatus, 295
  bMsisdn, 293
  bNia, 296
  bOCIOCT, 292
  bOperatorControlledPlmnSelectorAccessTech, 295
  bOperatorPlmnList, 295
  bPlmnNetworkName, 295
  bRfu1, 293
  bRplmnLastUsedAccessTech, 295
  bRunAtCommand, 294
  bSdn, 291
  bServiceProviderDisplayInfo, 295
  bShallBeSetToOne, 294
  bSms, 292
  bSmsCbdDataDownload, 293
  bSMSP, 292
  bSmsPpDataDownload, 293

  bSmsr, 292
  bSolsa, 293
  bSpn, 293
  bUserControlledPLMNSelectorAccessTech, 293
  bUsimCallControl, 294
  bUsimGpsCallControl, 296
  bUsimMoSmsControl, 294
  bVbsGroupId, 296
  bVgcsIdList, 296
TelUtility.h, 327
TimeZone
  TelNetworkTimeAndZone_t, 226
Ton
  TelSmsAddressInfo_t, 267
TotalCount
  TelSmsStoredMsgCountInfo_t, 274
TotalRecordCount
  TelSimPbStorageInfo_t, 248
TpDataCodingScheme
  TelSmsParams_t, 273
TpDestAddr
  TelSmsParams_t, 272
TpProtocolId
  TelSmsParams_t, 273
TpSvcCntrAddr
  TelSmsParams_t, 273
TpValidityPeriod
  TelSmsParams_t, 273
TunnelDataLen
  TelTapiTunnelInfo_t, 287
Type
  TelSsCallBarringInfo_t, 275
  TelSsForwardInfo_t, 280
TYPE_SS_TYPE_USSD
  SSDS, 206

UeccInfo
  TelSimEccData_t, 233
UpinStatus
  TelSimStatusData_t, 262
UsedCount
  TelSmsStoredMsgCountInfo_t, 274
UsedRecordCount
  TelSimPbStorageInfo_t, 248
UsimSst
  TelSimUstInfo_t, 266
UsimSstLen
  TelSimUstInfo_t, 266
UssdStringLength
  TelSsUssdMsgInfo_t, 284
UstData
  TelSimUstInfo_t, 266
UstInfo
  TelSimDeviceContextData_t, 232
Utility Data Structures, 207
UtilityDS
  TAPI_API_DEPRICATED, 207
  TAPI_API_INVALID_CALL_HANDLE, 208
  TAPI_API_INVALID_CALL_STATE, 207
  TAPI_API_INVALID_DATA_LEN, 208
  TAPI_API_INVALID_INPUT, 207
  TAPI_API_INVALID_PTR, 207
  TAPI_API_MODEM_POWERED_OFF, 208
  TAPI_API_NOT_SUPPORTED, 207
  TAPI_API_OEM_PLUGIN_FAILURE, 208
  TAPI_API_OFFLINE_MODE_ERROR, 208
  TAPI_API_REQUEST_MAX_IN_PROGRESS, 208
  TAPI_API_SERVER_FAILURE, 208
  TAPI_API_SERVICE_NOT_READY, 208
  TAPI_API_SIM_BUSY, 208
  TAPI_API_SIM_CARD_PERMANENT_ERROR, 208
  TAPI_API_SIM_INVALID_CARD_TYPE, 208
  TAPI_API_SIM_NOT_FOUND, 208
  TAPI_API_SIM_NOT_INITIALIZED, 208
  TAPI_API_SUCCESS, 207
  TAPI_API_SYSTEM_OUT_OF_MEM, 207
  TAPI_API_SYSTEM_RPC_LINK_DOWN, 207
  TAPI_EVENT_CLASS_UNKNOWN, 208
  TAPI_EVENT_UNKNOWN, 208
  TAPI_REGISTRATION_FAILURE, 208
UtilityDS
  TapiResult_t, 207
UusData
  TelCallUusInfo_t, 219
UusDataLen
  TelCallUusInfo_t, 219
UusInfo
  TelCallIncomingCallInfo_t, 212
  TelCallUserDataInfo_t, 218
UusType
  TelCallUusInfo_t, 219

WaitingRecord
  TelSsRecord_t, 283

Year
  TelNetworkTimeAndZone_t, 226